

## UTILITIES MODULE

### TECHNICAL REQUIREMENTS

#### *Introduction*

This document is the technical response to the SWSS Foster Care User Requirements for the Utilities Module. It will describe how the development team will implement the changes and additions to SWSS Childrens to effect the requirements.

This document is to be used as a tool by the development team when coding the solution or maintaining it in the future. Thus this document is likely to be updated during the lifecycle of the SWSS FAJ project. Versions of this document will be maintained in PVCS Version Manager, and the reader should be aware that multiple printed versions might exist.

#### *Module Description*

This module is used to maintain active SWSS FAJ user records in the SWSS FAJ Oracle database. More specifically, this module adds, assigns, and inactivates load numbers. User records may be added, changed, and inactivated (load numbers are never deleted!) in this module. The **CPS Transfer Information** module, the **Managing Related Cases** module, the **Payment Document Control Utility**, and the **Payments Online! Module** may also be accessed from here. This module also has the ability to display a list of inactive load numbers that have been used in the past and are currently in the Oracle database.

#### *Requirements*

##### Process Description

This module is mostly used to assign load numbers to valid SWSS FAJ users. This module can switch these load number assignments around and also inactivate them. The Utilities module allows users to be given security levels that will restrict their access to SWSS FAJ functions in appropriate ways. This "profiling" of SWSS FAJ users also allows the definition of worker-to-supervisor relationships. The creation and maintenance of these profiles allows valid users to gain access to the SWSS FAJ application. These valid users can then use the SWSS FAJ application to maintain vital data for Child Protective Services, Child Foster Care, Adoption, and Juvenile Justice cases. The Utilities module allows profiles to be designated as clerical workers, SWSS FAJ workers, or SWSS FAJ supervisors by means of a security level. Payment Document Control (P/DC) Utility users are also created and designated by a security level. This module can also give users the rights to manage Provider information or access the **Payment Document Control Utility**. The P/DC users will also have access to the **Payments Online! Module** by selecting an action labelled as **P/DC Modification**.

The Utilities module provides functions for creating and maintaining special Security Coordinator profiles. The list of possible county-district combinations that coordinators can cover is defined here. These coordinators are then responsible for maintaining valid and accurate user profile records.

The Utilities module provides a mechanism by which all valid SWSS FAJ users may change their own signon passwords. This function is also provided in such a way that Security Coordinators may reset the password of any users he/she is allowed to "cover". This module does not maintain Central Office Security Coordinator records. However, a recent change now allows Central Office Security Coordinators the ability to change their own passwords. A special exception to this is the CENT\_ADM username. The password for this username must never be changed and it must remain secret.

---

This module provides a mechanism for assigning county coverage to County Adoption Supervisors.

The Utilities module provides a tool to allow security coordinators the ability to view load numbers that have been added to the SWSS FAJ system, but have been inactivated. The list currently available to the coordinators only lists load numbers that have been inactivated and have not been assigned to another active worker. It also displays a complete list of the load number's history. This history lets a coordinator determine whom an inactive load number was assigned to and during what time period(s).

Finally, this module gives valid users access to the **CPS Transfer Information** module, the **Managing Related Cases** module, the **Payment Document Control Utility**, and the **Payments Online!** module.

## Functional Requirements

- 1.) Special code and screen configurations have been developed that will allow this module to act as the Staff Utilities screen for the StaffProfiles application. When the Tag property of the frmAddStaff form is set to "Staff Profiles", the Utilities module is configured for StaffProfiles.
- 1.1.) While in this "mode", the Application combo box does not allow access to the SWSS selection unless the user has selected the Change Staff Profile action. Even then, the user will only be able to change the password for his/her own profile.
- 2.) When this module is in the SWSS Utilities mode, Security Coordinator users are allowed to create and maintain LOA2 profile records. As a matter of fact, any valid SWSS profile can have his/her load numbers designated as both LOA2 and SWSS. A flag in the database designates a load number as either SWSS ("S"), LOA2 ("L"), or both ("B").
- 3.) Whenever this module creates a new staff profile, the Synchronize Synonyms process (SWSS.SWSS\_SET\_DEF\_SCHEMA\_PKGS) must be run.
- 4.) If a user is changing his/her own password, the password change must be encrypted and stored into the SWSS\_INI.INI file as well as changing it for his/her Oracle account in the database.
- 5.) The Utilities process is responsible for maintaining the proper assignments of Oracle Roles to most of the SWSS Oracle user accounts. SWSS profiles (and SWSS/LOA2 profiles) are the only ones which have Oracle user accounts on the database. Load numbers, which are LOA2 (only) load numbers, are not assigned to an Oracle user account. If a profile is set up as a Security Administrator, then that user's Oracle account must be assigned to the SWSS\_FIELD\_ADMIN role. Any other SWSS (or SWSS/LOA2) profile, created by the SWSS Utilities module, has the SWSS\_WORKER role assigned to the Oracle account. A third role, SWSS\_CENTRAL\_ADMIN, is assigned to the special "Central Office Security Coordinator" user(s). This special type of user does not require a load number and is, therefore, not maintained by the SWSS application. The Oracle database team carefully maintains this special user, his/her Oracle account and role assignments, and his/her database records.
- 6.) When changing passwords, users must begin passwords with a letter (A-Z). The first character of a password may no longer be a number. This is to prevent users from entering eight digits of their load number as a password. Unfortunately, Oracle does not support passwords that are completely numeric.

## Business Events

The following are business events that occur which will initiate use of the features in SWSS FAJ to add, change, or inactivate a SWSS load number.

- 1.) A county FIA field worker must be given access to the SWSS Utilities module for the purpose of creating and maintaining valid SWSS user profiles within a defined set of districts within one or more counties.
- 2.) A new Child Foster Care, Child Protective Services, Adoption, and/or Juvenile Justice worker has been hired into a county office. The new worker requires access to the SWSS FAJ application in order to maintain data for his/her caseload.
- 3.) A currently active worker's responsibilities at FIA have changed. The worker's profile needs a different set of programs he/she can access, a different security level, a different supervisor, etc.
- 4.) If a supervisor reverts back to being a worker, leaves a county office, or if supervisor assignments are switched around (perhaps to different units), all profiles that will be affected by this change must be modified individually.
- 5.) If currently active workers leave the county office, their profiles must be inactivated.

July 06, 2001

- 6.) If a load number must be transferred from one worker to another, the original profile with the load number must first be inactivated. Then the new, or receiving, worker/profile must be assigned to the inactivated load number. All active cases assigned to the transferred load number are effectively transferred to the new profile.
- 7.) If a SWSS worker being added to the system already has a matching LOA2 load number profile in the system, the security coordinator will be prompted to convert the existing load number into an LOA2/SWSS load number. Upon the successful completion of such a "conversion", the user must then make certain that any relevant ASSIST information is updated through the ASSIST application.
- 8.) If an active worker changes his/her marital status or his/her name is changed, a security coordinator can modify the name fields in the worker's profile. However, if the worker's username changes because of marital status or a name change, then the current profile must be inactivated. A new profile with the new username must then be created. The worker's original load number can then be assigned to the new username.
- 9.) A SWSS FAJ user has forgotten his/her password. A security coordinator can access the user's profile and change the password to a new value without needing to know the user's current password setting.
- 10.) A user believes that someone is covertly accessing his/her case information by logging into SWSS FAJ using his/her username and password. Or, a user wishes to keep his/her username a secret and decides to change his/her password more often than every ninety days. A user may access his/her own worker profile in the Utilities module and change his/her password. To verify that the correct user is changing the password, the user must enter his/her current password before entering a new password.

The following are business events that occur which will initiate use of the features in SWSS FAJ to display inactive load numbers.

- 1.) An investigation by the Central Office Security department requires data pertaining to a particular load number used by an inactive worker. As long as that load number has not been assigned to someone else, this tool can provide a complete history of a load number's assignment(s).

The following are business events that occur which will initiate use of the features in SWSS FAJ to access the **Payment Document Control Utility** or the **P/DC Modification** action (which will lead to the **Payments Online!** SWSS FAJ module).

- 1.) A PROGRAM OFFICE OR P/DC user wants to request a board and care payment authorization for a child.
- 2.) A PROGRAM OFFICE OR P/DC user wants to request a board and care payment for a child to be ended.
- 3.) A PROGRAM OFFICE OR P/DC user wants to request a board and care payment authorization for a child to be modified.
- 4.) A child's case situation has changed or a payment authorization mistake has been discovered. A PROGRAM OFFICE OR P/DC user must get into the **Payments Online!** module in order to correct payment authorization data.

The following are business events that occur which will initiate use of the features in SWSS FAJ to access the **CPS Transfer Information** process.

- 1.) On an unregistered SWSS FAJ case that came from a Conversion run, a "pointer" to some non-CIS case information may be waiting in a special database table {swss\_conversion} for a case worker. After selecting this unregistered case from Main Menu, Find A Client (Soundex), or from Case Listing, the user may execute this process from the Utilities module. If data exists in the {swss\_conversion} table for this case, this process will then allow the user to incorporate, into the case, the MAPPER data that this table's record "points" to. This is a substitute for having the user re-enter this non-CIS data manually and to provide a consistent starting point from MAPPER data that was entered while the case was a Child Protective Services case. Some of this information can include Five Day Packet, Legal, Funding Determination, companion, and Placement records, perhaps even more.

The following are business events that occur which will initiate use of the features in SWSS FAJ to access the **Manging Related Cases** module.

- 1.) A user needs to add a child to a companion group or...
- 2.) A user needs to remove a child from a companion case because:
  - 2.1.) A child's case closes (and other companions do not)

- 2.2.) A child is adopted (and the others are not)
- 2.3.) A child moves to another county and is assigned to a different load number (and the others do not)
- 2.4.) A link to another case is appropriate and needed.

## System Flow

See the *Utilities Technical Flow and Images.doc* document for more information.

## List of Program Units

This would be stuff like common VB code called, the number of VB .BAS modules and form modules in the current application. Also list the Stored Procedures called. Show a “Structure Diagram” of which VB procedure calls a stored procedure or another VB procedure. Also show which stored procedures call other stored procedures.

## Visual Basic Code Modules

<u>Forms</u>	<u>File Name</u>
frmAboutCFC	(Aboutcfc.frm)
frmError	(Error.frm)
frmSplash	(frmSplash.frm)
frmAddStaff	(ADDSTAFF.FRM)
frmInactiveLoadNumbers	(InactiveLoadNumbers.frm)
frmPDC_ModCaseSelect	(frmPDC_ModCaseSelect.frm)
frmSupervisorReports	(SupervisorReports.frm)

<u>Code Modules</u>	<u>File Name</u>
millerm7	(millerm7.bas)
swss	(swss.bas)
Utilities	(Utilities.bas)

<u>Class Modules</u>	<u>File Name</u>
clsConn	(ADOConnection.cls)

## SQL Stored Procedures and Packages

{SWSS.CENTRAL_SECURITY_ADMIN}	(D/B team package)
{SWSS.LOCAL_SECURITY_ADMIN}	(D/B team package)
{SWSS.IDENTIFY_COORD}	(D/B team package)
{SWSS.FIND_DB_USER}	(D/B team funtion)
{SWSS.ALTER_SWSS_USER}	(D/B team procedure)
{SWSS.PACKLISTS}	
{SWSS.UTILITIES}	
{SWSS.CENTRAL_OFFICE}	(D/B team package)
{SWSS.WORKER_MOD}	
{SWSS.RECONPKG}	

## Structure Diagrams

**Note:** For all references to frmError.Form\_Load in these tables, please refer to the Form\_Load entry of the **FIRST CALL DEPTH** column in the “**MODULE:** Error.frm” table in the *Common Elements Technical Requirements.doc* document.

July 06, 2001

<b>MODULE: UTILITIES.BAS</b>		
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>
Main	frmSplash.Form_Load	
	swss.CenterControl	
	swss.PrivateINI_Register	
	swss.PrivateINI_DeleteSection	
	swss.ExtractINI_Info	
	clsConn. Class_Initialize	
	frmAddStaff.Form_Load	(see the Form_Load entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table for more details)
	frmSplash.Form_Unload	

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
Form_Load	swss.PreviousInstance		
	swss.ExtractINI_Info		
	swss.SetEnabledProperty		
	fSetActionPanelVisible	See the fSetActionPanelVisible entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	swss.SetVisibleProperty		
	swss.CenterControl		
	swss.SetDeviceIndependentWindow		
	fMenuItemsEnabled	See the fMenuItemsEnabled entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	fEnabledFrameFields	See the fEnabledFrameFields entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	swss.IncrementMessageDots		
	fGetUserCoDist	See the fGetUserCoDist entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	cboActionCounty_Click	fCountiesIntoComboBox	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
Form_QueryUnload	Form_Unload	If user is a security administrator or a PDC user, then →	swss.CleanUpObjects
			swss.DeleteINI_Files
		Otherwise, mnuScreensMainMenu_Click	See the Common Elements technical documentation for the code under the mnuScreensMainMenu_Click Control Event
optAddChangeDelete_Click	fActionOptionEvent		
	If an error occurs, then frmError.Form_Load		
optAddChangeDelete_GotFocus	fActionOptionEvent		
cboActionCounty_KeyPress	swss.iFlexibleSelectItem		
cboActionCounty_Click	fCountiesIntoComboBox	See the fCountiesIntoComboBox entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	If an error occurs, then frmError.Form_Load		
cmdLeaveUtilities_Click	If the <i>record-changed</i> flag has been set to True, then →	mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column later in this table.
	Otherwise, Form_QueryUnload	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
mnuFileExit_Click	swss.change_process If the user does not select the Cancel button from the dialog box then, Form_QueryUnload	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
cmdChooseAction_Click	swss.SetEnabledProperty		
	swss.SetVisibleProperty		

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	If this is the first time the ChooseAction button has been selected and the user selected any option other than “CPS Transfer Information”, “Manage Companion Cases” or “Payment Document Control Utility”, then fOneTimeInitialization	See the fOneTimeInitialization entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	Else if the “Payment Document Control Utility” option was selected, then swss.LaunchEXE		See the Form_Load entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: frmPDC_ModCaseSelect.FRM</b> (Control Events)” table later in this document.
	Else if the “P/DC Modification” action was selected, then →	frmPDC_ModCaseSelect.Form_Load	
	Else if the “Manage Companion Cases” option was selected, then →	swss.PrivateINI_Register	
		swss.tGetAppPath	
		swss.lPrivateINI_PutString	
		swss.CommonShell	
		Leave the cmdChooseAction_Click subroutine. No other code in this subroutine will be called.	
	Else if the “CPS Transfer Information” option was selected, then →	If the user has selected a case to which he/she has full update access, then fReconciliation	See the fReconciliation entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.
		Else if the user did not select a case before entering the Utilities module and wishes to chose one from the Case Listing module, then swss.tOfficialModuleName and mnuScreensCaseListing_Click	See the Common Elements technical documentation for the code under the mnuScreensCaseListing_Click Control Event
		Leave the cmdChooseAction_Click subroutine. No other code in this subroutine will be called.	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	If the user has not selected the “Display Inactive Load Numbers” action, then →	fSetActionPanelVisible	See the fSetActionPanelVisible entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.
		fSetWorkerPanelA_Visible	See the fSetWorkerPanelA_Visible entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.
	If the user is a security administrator then →	If the user has selected the “Add Staff Profile” option, then fInitializeAddMode	See the fInitializeAddMode entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.
		Else if the user has selected the “Change Staff Profile” option, then fInitializeEditMode	See the fInitializeEditMode entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.
		Else if the user selected the “Inactivate Staff Profile” option, then fInitializeEditMode and fEnableOrDisableAllDataFields	See the fInitializeEditMode and fEnableOrDisableAllDataFields entries in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.
		Else if the user selects the “Display Inactive Load Numbers” option, then frmInactiveLoadNumbers.Form_Load	See the Form_Load entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: InactiveLoadNumbers.FRM</b> (Control Events)” table later in this document.



July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	Else →	fInitializeEditMode	See the fInitializeEditMode entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.
	If the user has not selected the “Display Inactive Load Numbers” action, then →	fSetHeaderLabel	See the fSetHeaderLabel entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.
	fUserFieldAccess	See the fUserFieldAccess entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	cboLoadNumberList_Got Focus		
	swss.bSaferSetFocus		
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
cboLoadNumberList_Click	fClearDataFields	See the fClearDataFields entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	fGetWorkerRecord	See the fGetWorkerRecord entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	swss.SetVisibleProperty		
	millerm7.bUserAlreadyLoggedIn		
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
	swss.iFlexibleSelectItem		
cboLoadNumberList_KeyPress			

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
cboCountyCodeListBox_Click	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	fSelectedDistrict	See the fSelectedDistrict entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	fGetCountyDependentWorkers	See the fGetCountyDependentWorkers entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
cboCountyCodeListBox_KeyPress	swss.iFlexibleSelectItem		
cboCountyCodeListBox_LostFocus	fWorkerAccessNotice	See the fWorkerAccessNotice entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
cboAlternateList_Key Down	If an error occurs, then →	frmError.Form_Load	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		mnuFileExit_Click	
cboAlternateList_Key Press	swss.iFlexibleSelectItem		
cboAlternateList_Lost Focus	swss.ClearCollection		
	swss.lBrakeOutElements		
	swss.BreakUpLoadNumber		
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
cboApplication_Click	fDetermineApplicationMode	See the fDetermineApplicationMode entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	cmdChooseAction_Click	See cmdChooseAction_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	swss.SetEnabledProperty		
	fSetHeaderLabel	See the fSetHeaderLabel entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
cboDistrict_KeyPress	swss.iFlexibleSelectItem		
cboDistrict_LostFocus	fSelectedDistrict	See the fSelectedDistrict entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
cboLanguages_Click	cboLanguages_LostFocus	See cboLanguages_LostFocus entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
cboLanguages_KeyDown	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
cboLanguages_LostFocus	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
cboSecurityLevel_Click	cboSecurityLevel_Click (A recursive call that does not seem to overflow the stack)		

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	cboLoadNumberList_Click	See cboLoadNumberList_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fDetermineApplicationMode	See the fDetermineApplicationMode entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	fADPT_SpvsrCountiesEnabled	See the fADPT_SpvsrCountiesEnabled entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	fUserFieldAccess	See the fUserFieldAccess entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
cboSecurityLevel_KeyPress	swss.iFlexibleSelectItem		
cboSecurityLevel_LostFocus	cboSecurityLevel_Click	See cboSecurityLevel_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	If an error occurs, then frmError.Form_Load		
cboSupervisorList_KeyDown	swss.bSaferSetFocus		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
cboSupervisorList_KeyPress	swss.iFlexibleSelectItem		
cboSupervisorList_LostFocus	swss.ClearCollection		
	swss.lBrakeOutElements		

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	fbVerifySupervisor	See the fbVerifySupervisor entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	swss.BreakUpLoadNumber		
	swss.bSaferSetFocus		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
chkAccess_Click	fCoordinatorCountiesEnabled	See the fCoordinatorCountiesEnabled entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	cboAlternateList_KeyDown	See cboAlternateList_KeyDown entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	cboAlternateList_LostFocus	See cboAlternateList_LostFocus entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	cboSupervisorList_KeyDown	See cboSupervisorList_KeyDown entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	cboSupervisorList_LostFocus	See cboSupervisorList_LostFocus entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	swss.SetEnabledProperty		
	chkLOA_Coord_Click	See chkLOA_Coord_Click entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	cmdRemoveCounty_Click	See cmdRemoveCounty_Click entry in the <b>FIRST CALL DEPTH</b> column later in this table.	

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	ftGetUpdateFlag	See the ftGetUpdateFlag entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	fADPT_SpvSrCountiesEnabled	See the fADPT_SpvSrCountiesEnabled entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	cmdRemoveADPT_Count y_Click	See cmdRemoveADPT_Count_Click entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	fbUsernameHasManyLoadNumbers	See the fbUsernameHasManyLoadNumbers entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	chkAccess_Click (a recursive call which does not seem to fill the call stack)		
	fUserFieldAccess	See the fUserFieldAccess entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
chkAccess_LostFocus	swss.ReplaceString		
	cmdAddADPT_Count_Click		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
chkLOA_Coord_Click	fSetDataPanelCaptions		
	swss.SetEnabledProperty		
	If an error occurs, then frmError.Form_Load		

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
cmdAddADPT_Count y_Click	If an error occurs, then →	frmError.Form_Load	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		mnuFileExit_Click	
cmdAddCounty_Click	fbDistrictValidity	See the fbDistrictValidity entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Procedures)” table later in this document.	
	swss.SetEnabledProperty		
	If an error occurs, then →	frmError.Form_Load	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		mnuFileExit_Click	
cmdAddressCopy_Cli ck	If an error occurs, then →	frmError.Form_Load	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		mnuFileExit_Click	
cmdAddressCoToMail ing_Click	fGetCountyAddress	See the fGetCountyAddress entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Procedures)” table later in this document.	
	txtAddressLine1_LostFoc us	See txtAddressLine1_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtAddressLine2_LostFoc us	See txtAddressLine2_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtCity_LostFocus	See txtCity_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtZipcode_LostFocus	See txtZipcode_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtZipPlus_LostFocus	See txtZipPlus_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
cmdAddressCoToPri mary_Click	fGetCountyAddress	See the fGetCountyAddress entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Procedures)” table later in this document.	

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	txtAddressLine1_LostFocus	See txtAddressLine1_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtAddressLine2_LostFocus	See txtAddressLine2_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtCity_LostFocus	See txtCity_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtZipcode_LostFocus	See txtZipcode_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtZipPlus_LostFocus	See txtZipPlus_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
cmdAddressDistToMailing_Click	fGetDistrictAddress	See the fGetDistrictAddress entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	txtAddressLine1_LostFocus	See txtAddressLine1_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtAddressLine2_LostFocus	See txtAddressLine2_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtCity_LostFocus	See txtCity_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtZipcode_LostFocus	See txtZipcode_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtZipPlus_LostFocus	See txtZipPlus_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
cmdAddressDistToPrimary_Click	fGetDistrictAddress	See the fGetDistrictAddress entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	txtAddressLine1_LostFocus	See txtAddressLine1_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	



<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	txtAddressLine2_LostFocus	See txtAddressLine2_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtCity_LostFocus	See txtCity_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtZipcode_LostFocus	See txtZipcode_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtZipPlus_LostFocus	See txtZipPlus_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
cmdCancel_Click	swss.SetEnabledProperty		
	swss.SetVisibleProperty		
	fSetActionPanelVisible	See the fSetActionPanelVisible entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	fSetWorkerPanelC_Visible	See the fSetWorkerPanelC_Visible entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	fSetWorkerPanelB_Visible	See the fSetWorkerPanelB_Visible entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	fSetWorkerPanelA_Visible	See the fSetWorkerPanelA_Visible entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Procedures)</b> ” table later in this document.	
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
cmdCentPasswordCancel_Click	swss.SetVisibleProperty		
	swss.SetEnabledProperty		

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	fSetActionPanelVisible	See the fSetActionPanelVisible entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	If an error occurs, then frmError.Form_Load		
cmdCentPasswordOK_Click	swss.bSaferSetFocus		
	cmdCentPasswordCancel_Click	See the cmdCentPasswordCancel_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	clsConn.Execute	See the Execute entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADOConnection.cls</b> (Methods)” table later in this document.	
	{SWSS.CENTRAL_SECURITY_ADMIN.MODIFY_SWSS_USER}		
	swss.PrivateINI_Register		
	swss.tGetAppPath		
	swss.lPrivateINI_PutString		
	swss.tEncryptString		
	clsConn.Class_Terminate	See the Class_Terminate entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADOConnection.cls</b> (Events)” table later in this document.	
	clsConn.Class_Initialize	See the Class_Initialize entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADOConnection.cls</b> (Events)” table later in this document.	
	If an error occurred then →	swss.bSaferSetFocus swss.DisplayADO_Errors frmError.Form_Load	
cmdContinue_Click	swss.SetEnabledProperty		
	swss.SetVisibleProperty		
	mshPhone_LostFocus		

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	mskPhoneExtension_LostFocus	See the mskPhoneExtension_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtEmail_LostFocus	See the txtEmail_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtWorkerNameLast_LostFocus	See the txtWorkerNameLast_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtWorkerNameSuffix_LostFocus	See the txtWorkerNameSuffix_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtWorkerNameFirst_LostFocus	See the txtWorkerNameFirst_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtWorkerNameMiddle_LostFocus	See the txtWorkerNameMiddle_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	txtUsername_LostFocus	See the txtUsername_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	cboCountyCodeListBox_LostFocus	See the cboCountyCodeListBox_LostFocus entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	cboDistrict_LostFocus	See the cboDistrict_LostFocus entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	mskSection_LostFocus	See the mskSection_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	mskUnit_LostFocus	See the mskUnit_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	mskWorkerNumber_LostFocus	See the mskWorkerNumber_LostFocus entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	fbCheckSSN	See the fbCheckSSN entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	fbNameFieldsError	See the fbNameFieldsError entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	fbLoadNumberError	See the fbLoadNumberError entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	fbPhoneFieldError	See the fbPhoneFieldError entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	cboSecurityLevel_LostFocus		
	fbSecurityChanged	See the fbSecurityChanged entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	fbUsernameHasManyLoadNumbers	See the fbUsernameHasManyLoadNumbers entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	frmSupervisorReports.Form_Load	See the Form_Load entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: SupervisorReports.FRM</b> (Control Events)” table later in this document.	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	fSetDataPanelCaptions	See the fSetDataPanelCaptions entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	fbSaveWorkerData	See the fbSaveWorkerData entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	fSetWorkerPanelB_Visible	See the fSetWorkerPanelB_Visible entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	fbCheckForSupervisor	See the fbCheckForSupervisor entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	fSetWorkerPanelC_Visible	See the fSetWorkerPanelC_Visible entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	swss.bSaferSetFocus		
	fbPhoneFieldError	See the fbPhoneFieldError entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
cmdRemoveADPT_County_Click	swss.SetEnabledProperty		
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
cmdRemoveCounty_Click	swss.SetEnabledProperty		
	If an error occurs, then	frmError.Form_Load	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	→	mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
lstADPT_CountiesAssigned_DblClick	cmdRemoveADPT_Count_y_Click	See cmdRemoveADPT_County_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
lstADPT_CountiesPool_DblClick	cmdAddADPT_County_Click	See cmdAddADPT_County_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
lstCountyPool_Click	swss.SetEnabledProperty		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
lstDistrictPool_DblClick	cmdAddCounty_Click	See cmdAddCounty_County_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
mnuPassword_Click	fSetActionPanelVisible	See the fSetActionPanelVisible entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	swss.SetVisibleProperty		
	swss.bSaferSetFocus		
	If an error occurs, then →	frmError.Form_Load	
mskPhone_GotFocus	swss.select_text		
	If an error occurs, then →	frmError.Form_Load	
mskPhone_KeyDown	mskPhone_LostFocus		
	cmdContinue_Click	See cmdContinue_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
mskPhone_Validation Error	mskPhone_LostFocus		
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
mskPhoneExtension_GotFocus	swss.select_text		
mskPhoneExtension_	If an error occurs, then	frmError.Form_Load	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
LostFocus	→	mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
mskSection_GotFocus	swss.select_text		
mskSection_LostFocus	fWorkerAccessNotice	See the fWorkerAccessNotice entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
mskSSN_GotFocus	swss.select_text		
mskSSN_LostFocus	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
mskUnit_GotFocus	swss.select_text		
mskUnit_LostFocus	fWorkerAccessNotice	See the fWorkerAccessNotice entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
mskWorkerNumber_GotFocus	swss.select_text		
mskWorkerNumber_LostFocus	fWorkerAccessNotice	See the fWorkerAccessNotice entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Procedures)” table later in this document.	
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtAddressLine1_Key	If an error occurs, then	frmError.Form_Load	

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
Up	→	mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtAddressLine1_Lost Focus	If an error occurs, then →	frmError.Form_Load	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		mnuFileExit_Click	
txtAddressLine2_Key Up	If an error occurs, then →	frmError.Form_Load	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		mnuFileExit_Click	
txtAddressLine2_Lost Focus	If an error occurs, then →	frmError.Form_Load	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		mnuFileExit_Click	
txtCentPasswordConfirm_GotFocus	swss.select_text		
txtCentPasswordConfirm_KeyPress	millerm7.i1stUNamePwdChar		
	millerm7.iUNamePwdCharacter		
txtCentPasswordConfirm_LostFocus	millerm7.bPasswordMinLength		
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtCentPasswordNew_GotFocus	swss.select_text		
txtCentPasswordNew_KeyPress	millerm7.i1stUNamePwdChar		
	millerm7.iUNamePwdCharacter		
txtCentPasswordNew_LostFocus	millerm7.bPasswordMinLength		
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtCity_KeyUp	If an error occurs, then →	frmError.Form_Load	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		mnuFileExit_Click	
txtCity_LostFocus	If an error occurs, then	frmError.Form_Load	



<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	→	mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtEMail_LostFocus	If an error occurs, then →	frmError.Form_Load	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		mnuFileExit_Click	
txtPasswordConfirm_GotFocus	swss.select_text		
txtPasswordConfirm_KeyPress	millerm7.i1stUNamePwdChar		
	millerm7.iUNamePwdCharacter		
txtPasswordConfirm_LostFocus	millerm7.bPasswordMinLength		
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtPasswordNew_GotFocus	swss.select_text		
txtPasswordNew_KeyPress	millerm7.i1stUNamePwdChar		
	millerm7.iUNamePwdCharacter		
txtPasswordNew_LostFocus	millerm7.bPasswordMinLength		
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtPasswordOld_GotFocus	swss.select_text		
txtPasswordOld_KeyPress	millerm7.iUNamePwdCharacter		
txtUsername_KeyPress	millerm7.i1stUNamePwdChar		
	millerm7.iUNamePwdCharacter		
	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtUsername_LostFocus	swss.bSaferSetFocus		
	If an error occurs, then	frmError.Form_Load	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Control Events)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	→	mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtWorkerNameFirst_LostFocus	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtWorkerNameLast_LostFocus	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtWorkerNameMiddle_LostFocus	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtWorkerNameSuffix_LostFocus	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtZipcode_KeyUp	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtZipcode_LostFocus	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
txtZipPlus_KeyUp	If an error occurs, then →	frmError.Form_Load mnuFileExit_Click	See mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
ClearWorkerUDT	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
			<b>“MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.
fActionOptionEvent	If an error occurs, then →	frmError.Form_Load	
fADPT_SpvsrCounties Enabled	swss.SetEnabledProperty		
	cmdRemoveADPT_Count y_Click	See the cmdRemoveADPT_County_Clic k entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.
fbAddNewWorker	millerm7.tBuildSQLStrin g		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Proce dure		
	{SWSS.UTILITIES.oracle_ account_exists}		
	If the {SWSS.UTILITIES} package is invalid, then exit the fbAddNewWorker procedure.		
	swss.FillQueriedResultse t		
	{SWSS.CENTRAL_SECU RITY_ADMIN.CREATE_L CL_SEC_COORD}		
	If the {SWSS.CENTRAL_SECU RITY_ADMIN} package is invalid, then exit the fbAddNewWorker procedure.		

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	{SWSS.LOCAL_SECURITY_ADMIN.CREATE_SWS_WORKER}		
	If the {SWSS.LOCAL_SECURITY_ADMIN} package is invalid, then exit the fbAddNewWorker procedure.		
	If an error occurs, then →	swss.DisplayADO_Errors	
		frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fbAddressChange	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fbAdministerAddress	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.WORKER_MOD.save_worker_address}		
	If the {SWSS.WORKER_MOD} package is invalid, then exit the fbAdministerAddress procedure.		
	swss.IncrementMessageDots		
	If an error occurs, then →	swss.DisplayADO_Errors	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		frmError.Form_Load	
fbAdministerAdptSupvsrCo	millerm7.tBuildSQLString		
	millerm7.CallSQL_Procedure		
	{SWSS.WORKER_MOD.save_ADPT_sup_co}		
	If the {SWSS.WORKER_MOD} package is invalid, then exit the fbAdministerAdptSupvsrCo procedure.		
	{SWSS.WORKER_MOD.delete_ADPT_sup_co}		
	If the {SWSS.WORKER_MOD} package is invalid, then exit the fbAdministerAdptSupvsrCo procedure.		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fbAdministerCoord	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.WORKER_MOD.save_security_assign}		
	If the {SWSS.WORKER_MOD} package is invalid, then exit the fbAdministerCoord procedure.		
	If an error occurs, then →	swss.DisplayADO_Errors	
		frmError.Form_Load	
fbAdministerHours	millerm7.tBuildSQLString		

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	g		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.WORKER_MOD.save_worker_hrs }		
	If the {SWSS.WORKER_MOD} package is invalid, then exit the fbAdministerHours procedure.		
	swss.IncrementMessageDots		
	If an error occurs, then →	swss.DisplayADO_Errors	
		frmError.Form_Load	
fbAdministerLanguages	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.tBuildSQLString		
	millerm7.CallSQL_Procedure		
	{SWSS.WORKER_MOD.save_worker_languages}		
	If the {SWSS.WORKER_MOD} package is invalid, then exit the fbAdministerLanguages procedure.		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fbAdministerLoadNumbers	fbLoadChange	See the fbLoadChange entry in the <b>FIRST CALL DEPTH</b> later in this table.	
	swss.tBuildLoadNumber		
	fbDistrictValidity	See the fbDistrictValidity entry in the <b>FIRST CALL DEPTH</b>	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		later in this table.	
	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.WORKER_MOD.save_worker_load}		
	If the {SWSS.WORKER_MOD} package is invalid, then exit the fbAdministerLoadNumbers procedure.		
	If an error occurs, then →	swss.DisplayADO_Errors	
		frmError.Form_Load	
fbAdministerPhone	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.WORKER_MOD.save_worker_phone}		
	If the {SWSS.WORKER_MOD} package is invalid, then exit the fbAdministerPhone procedure.		
	swss.IncrementMessageDots		
	If an error occurs, then →	swss.DisplayADO_Errors	
		frmError.Form_Load	
fbAdministerRoles	fbModifyWorkerRoles	See the fbModifyWorkerRoles entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.bSynchronizeSynonyms		
	fbUsernameHasOtherPrograms	millerm7.tBuildSQLString	
		tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
			this table.
		millerm7.CallSQL_Procedure	
		{SWSS.UTILITIES.OtherAssgnd Progs}	
		If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
		If an error occurs, then →	frmError.Form_Load
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fbAdministerSecurity	fbSecurityChanged	See the fbSecurityChanged entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.FillQueriedResultset		
	{SWSS.WORKER_MOD.save_security_code}		
	If the {SWSS.WORKER_MOD} package is invalid, then exit the fbAdministerSecurity procedure.		
	fGetCountyDependentWorkers	See the fGetCountyDependentWorkers entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.IncrementMessageDots		
	If an error occurs, then →	swss.DisplayADO_Errors	



<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		frmError.Form_Load	
fbAdminLOA2LoadNo	fbLoadChange	See the fbLoadChange entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	fbDistrictValidity	See the fbDistrictValidity entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.WORKER_MOD.save_worker_load}		
	If the {SWSS.WORKER_MOD} package is invalid, then exit the fbAdminLOA2LoadNo procedure.		
	swss.IncrementMessageDots		
	If an error occurs, then →	swss.DisplayADO_Errors	
		frmError.Form_Load	
fbCheckForActiveCases	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.FillQueriedResultset		
	{SWSS.UTILITIES.get_active_cases}		
	If the { SWSS.UTILITIES } package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	swss.SetVisibleProperty		
	swss.SetEnabledProperty		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b>

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
			ADDSTAFF.FRM (Control Events)" table previously in this document.
fbCheckForActiveLoadNumbers	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.get_log_in_names}		
	If the { SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the "MODULE: ADDSTAFF.FRM (Control Events)" table previously in this document.	
	fbNonSWSS_Role	swss.ReplaceString	
		fbModifyWorkerRoles	See the fbModifyWorkerRoles entry in the <b>FIRST CALL DEPTH</b> column later in this table.
		If an error occurs, then →	frmError.Form_Load
			mnuFileExit_Click
	If an error occurs, then →	frmError.Form_Load	
fbCheckForLoginName	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	clsConn.Execute		
	{SWSS.UTILITIES.CheckForLoginName}		
	fbCheckForMatchingLoadNumber	millerm7.tBuildSQLString	
		tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.
		clsConn.Execute	
		{SWSS.UTILITIES.CheckLoadNoMatch}	
		If an error occurs, then →	frmError.Form_Load
			swss.SetEnabledProperty
	If an error occurs, then	frmError.Form_Load	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	→		
		swss.SetEnabledProperty	
fbCheckForSupervisor	swss.bSaferSetFocus		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fbCheckSSN	swss.bSaferSetFocus		
	swss.IncrementMessageDots		
	SSN_VALIDATE	If an error occurs, then →	frmError.Form_Load
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fbDistrictValidity	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fbDuplicateActiveLoad	swss.tBuildLoadNumber		
	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.duplic		

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	ate_active_load}		
	If the { SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fbErrorDuplicateLoad	millerm7.tBuildSQLStrin g		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	clsConn.Execute		
	{SWSS.WORKER_MOD.C onvertLOA_Load}		
	If an error occurs, then →	frmError.Form_Load	
		swss.SetEnabledProperty	
fbErrorSecurityEntry	swss.bSaferSetFocus		
	If an error occurs, then →	frmError.Form_Load	
fbInactivateWorker	swss.tBuildLoadNumber		
	fbCheckForActiveCases	See the fbCheckForActiveCases entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbCheckForActiveLoadNu mbers	See the fbCheckForActiveLoadNumbers entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	millerm7.tBuildSQLStrin g		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Proce dure		
	{SWSS.CENTRAL_SECU RITY_ADMIN.MODIFY_S		

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	WSS_USER }		
	{SWSS.LOCAL_SECURITY_ADMIN.MODIFY_SWS_WORKER}		
	If either the {SWSS.CENTRAL_SECURITY_ADMIN} package or the {SWSS.LOCAL_SECURITY_ADMIN} package is invalid, then exit the fbInactivateWorker procedure.		
	{SWSS.WORKER_MOD.inactivate_load_no}		
	If the {SWSS.WORKER_MOD} package is invalid, then exit the fbInactivateWorker procedure.		
	If an error occurs, then →	swss.DisplayADO_Errors	
		frmError.Form_Load	
fbInvalidConnection	If an InvalidConnection error is detected or if an error occurs, then →	frmError.Form_Load	
		swss.SetEnabledProperty	
		swss.CleanUpObjects	
		swss.ClearINI_Sections	
		Shut down the SWSS FAJ application.	
fbLoadChange	swss.tBuildLoadNumber		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fbLoadNumberError	fSetWorkerPanelA_Visible	See the fSetWorkerPanelA_Visible entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.IncrementMessage Dots		
	cboSecurityLevel_Click	See the cboSecurityLevel_Click	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fbModifyLOA2_Records	fbDuplicateActiveLoad	See the fbDuplicateActiveLoad entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbNameChange	See the fbNameChange entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.ReplaceString		
	clsConn.Execute		
	{SWSS.WORKER_MOD.save_LOA2_worker}		
	swss.IncrementMessageDots		
	swss.tBuildLoadNumber		
	fbAdminLOA2LoadNo	See the fbAdminLOA2LoadNo entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbAdministerPhone	See the fbAdministerPhone entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbAdministerHours	See the fbAdministerHours entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
			in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fbModifyWorkerRoles	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.FillQueriedResultset		
	{SWSS.CENTRAL_SECURITY_ADMIN.GRANT_REVOKE_ROLE}		
	{SWSS.LOCAL_SECURITY_ADMIN.GRANT_REVOKE_ROLE }		
	If either the {SWSS.CENTRAL_SECURITY_ADMIN} package or the {SWSS.LOCAL_SECURITY_ADMIN} package is invalid, then exit the fbModifyWorkerRoles procedure.		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fbModifyWorkerTable	fbModifyLOA2_Records	See the fbModifyLOA2 entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbUserEntryError	See the fbUserEntryError entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	fbDuplicateActiveLoad	See the fbDuplicateActiveLoad entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbAdministerRoles	See the fbAdministerRoles	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbNameChange	See the fbNameChange entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.ReplaceString		
	clsConn.Execute		
	{SWSS.WORKER_MOD.save_worker_name}		
	swss.IncrementMessageDots		
	swss.tBuildLoadNumber		
	fbAdministerLoadNumbers	See the fbAdministerLoadNumbers entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbAdministerSecurity	See the fbAdministerSecurity entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbAdministerAddress	See the fbAdministerAddress entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbAdministerPhone	See the fbAdministerPhone entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	swss.FillQueriedResultset		
	{SWSS.WORKER_MOD.save_load_program}		
	If the {SWSS.WORKER_MOD} package is invalid, then exit the fbModifyWorkerTable procedure.		
	fbAdministerLanguages	See the fbAdministerLanguages entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbAdministerCoord	See the fbAdministerCoord entry in the <b>FIRST CALL</b>	



July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		<b>DEPTH</b> column previously in this table.	
	fbAdministerAdptSupvsrCo	See the fbAdministerAdptSupvsrCo entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbAdministerHours	See the fbAdministerHours entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	{SWSS.CENTRAL_SECURITY_ADMIN.MODIFY_SWSS_USER }		
	{SWSS.LOCAL_SECURITY_ADMIN.MODIFY_SWSS_WORKER }		
	{ORADBA.ALTER_SWSS_USER }		
	swss.PrivateINI_Register		
	swss.tGetAppPath		
	swss.lPrivateINI_PutString		
	clsConn.Class_Terminate		
	cls.Class_Initialize		
	If an error occurs, then →	fSetWorkerPanelA_Visible	See the fSetWorkerPanelA_Visible entry in the <b>FIRST CALL DEPTH</b> column later in this table.
		swss.DisplayADO_Errors	
		frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fbNameChange	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
			document.
fbNameFieldsError	fSetWorkerPanelA_Visibl e	See the fSetWorkerPanelA_Visible entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.IncrementMessage Dots		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.
fbPhoneChange	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.
fbPhoneFieldError	swss.bSaferSetFocus		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.
fbSaveNewProfile	fbUserEntryError	See the fbUserEntryError entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	ftExistingOracleAccount	See the ftExistingOracleAccount entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.ReplaceString		
	swss.tBuildLoadNumber		
	ftProgramAccess	If an error occurs, then →	frmError.Form_Load

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
			Form_QueryUnload
	fbCheckForLoginName	See the fbCheckForLoginName entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	clsConn.Execute		
	{SWSS.WORKER_MOD.save_new_profile}		
	millerm7.Store_SQL_Time		
	If an error occurs, then →	fbErrorDuplicateLoad	See the fbErrorDuplicateLoad entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		frmError.Form_Load	
		Form_QueryUnload	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fbSaveWorkerData	millerm7.bUserAlreadyLoggedIn		
	swss.SetVisibleProperty		
	cboCountyCodeListBox_LostFocus	See the cboCountyCodeListBox_LostFocus entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	swss.tBuildLoadNumber		
	fbModifyLOA2_Records	See the fbModifyLOA2_Records entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbSaveNewProfile	See the fbSaveNewProfile entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbAdministerPhone	See the fbAdministerPhone entry in the <b>FIRST CALL DEPTH</b> column previously in	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		this table.	
	fbAdministerAdptSupvsrCo	See the fbAdministerAdptSupvsrCo entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbAdministerLanguages	See the fbAdministerLanguages entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbAdministerAddress	See the fbAdministerAddress entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbAdministerCoord	See the fbAdministerCoord entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbModifyWorkerTable	See the fbModifyWorkerTable entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fbInactivateWorker	See the fbInactivateWorker entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	cmdCancel_Click	See the cmdCancel_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.	
	fSetWorkerPanelA_Visible	See the fSetWorkerPanelA_Visible entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fbSecurityChanged	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
			in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fbUserEntryError	fSetWorkerPanelA_Visibl e	See the fSetWorkerPanelA_Visible entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.IncrementMessage Dots		
	swss.bSaferSetFocus		
	fSetWorkerPanelB_Visibl e	See the fSetWorkerPanelB_Visible entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fbUsernameHasMany LoadNumbers	millerm7.tBuildSQLStrin g		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Proce dure		
	{SWSS.UTILITIES.NoOfA ctiveLoadNos}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.	
	If an error occurs, then →	frmError.Form_Load	
fbVerifySupervisor	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	If an error occurs, then →	frmError.Form_Load	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fClearCountyDistrictInfo			
fClearDataFields	swss.SetEnabledProperty		
	cboSecurityLevel_Click	See the cboSecurityLevel_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.	
	fClearMaskedEditBox	See the fClearMaskedEditBox entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	chkLOA_Coord_Click	See the chkLOA_Coord_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fClearMaskedEditBox	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fCoordinatorCounties	swss.SetEnabledProperty		

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
Enabled			
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.
fCountiesIntoCombo Box	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.
fDetermineApplicatio nMode	swss.SetEnabledProperty		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.
fEnabledFrameFields	swss.SetEnabledProperty		
	fEnableFieldLabelPairs	swss.SetEnabledProperty	
		If an error occurs, then →	frmError.Form_Load
	fbUsernameHasManyLoa dNumbers	See the fbUsernameHasManyLoadNum bers entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	If an error occurs, then →	frmError.Form_Load	
fEnableOrDisableAllD ataFields	swss.SetEnabledProperty		
	fCoordinatorCountiesEn abled	See the fCoordinatorCountiesEnabled entry in the <b>FIRST CALL</b>	

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		<b>DEPTH</b> column previously in this table.	
	fADPT_SpvSrCountiesEnabled	See the fADPT_SpvSrCountiesEnabled entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fGetAllWorkers	swss.CallSQL_Procedure		
	{SWSS.UTILITIES.get_wrkr_load_no}		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.	
	swss.IncrementMessageDots		
	swss.tBuildLoadNumber		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fGetCountyAddress	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.tBuildSQLString		
	millerm7.CallSQL_Procedure		



July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	{SWSS.UTILITIES.get_county_address}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	fClearCountyDistrictInfo	See the fClearCountyDistrictInfo entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fGetCountyDependentWorkers	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.get_alt_workers}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	swss.IncrementMessageDots		
	swss.tBuildLoadNumber		
	swss.tBuildDelimitedString		
	{SWSS.UTILITIES.get_susp_workers}		
	swss.ClearCollection		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL</b>

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
			<b>DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fGetDistrictAddress	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.tBuildSQLString		
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.get_district_address}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.	
	fClearCountyDistrictInfo	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fGetLanguages	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.packlists.get_language}		
	If the {SWSS.packlists} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.	
	swss.IncrementMessage		

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	Dots		
	swss.tBuildDelimitedString		
	If an error occurs, then →	fbInvalidConnection	See the fbInvalidConnection entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fGetSecurityLevels	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.get_security_codes}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.	
	swss.IncrementMessage Dots		
	If an error occurs, then →	fbInvalidConnection	See the fbInvalidConnection entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fGetUserCoDist	millerm7.tBuildSQLString		

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	g		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.IncrementMessage Dots		
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.get_co_dist}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	fCountiesIntoComboBox	See the fCountiesIntoComboBox entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fGetWorkerAddress	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.get_worker_address}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	swss.IncrementMessage Dots		
	{SWSS.UTILITIES.IsSpecialAddress}		
	If an error occurs, then	fbInvalidConnection	See the

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	→		fbInvalidConnection entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fGetWorkerADPT_Spv srCo	swss.SetEnabledProperty		
	millerm7.tBuildSQLStrin g		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Proce dure		
	{SWSS.UTILITIES.get_AD PT_sup_co}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.	
	If an error occurs, then →	fbInvalidConnection	See the fbInvalidConnection entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fGetWorkerBasics	millerm7.tBuildSQLStrin g		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.get_worker_info}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	cboSecurityLevel_Click	See the cboSecurityLevel_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	swss.IncrementMessageDots		
	swss.BreakUpLoadNumber		
	swss.SetEnabledProperty		
	fGetCountyDependentWorkers	See the fGetCountyDependentWorkers entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	swss.tBuildLoadNumber		
	swss.ClearCollection		
	swss.lBrakeOutElements		
	swss.tBuildDelimitedString		
	If an error occurs, then →	fbInvalidConnection	See the fbInvalidConnection entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fGetWorkerCoDist	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.get_security_co_dist}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	If an error occurs, then →	fbInvalidConnection	See the fbInvalidConnection entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fGetWorkerHours	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.get_worker_hrs}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
fGetWorkerLanguages	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.get_worker_languages}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	swss.ClearCollection		
	swss.lBrakeOutElements		
	swss.IncrementMessageDots		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fGetWorkerLoadPrograms	swss.IncrementMessageDots		
	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.get_load_programs}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	swss.ClearCollection		
	swss.lBrakeOutElements		
	swss.IncrementMessageDots		



<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	swss.SetEnabledProperty		
	If an error occurs, then →	fbInvalidConnection	See the fbInvalidConnection entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fGetWorkerPhone	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.UTILITIES.get_worker_phone}		
	If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	fClearMaskedEditBox	See the fClearMaskedEditBox entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	If an error occurs, then →	fbInvalidConnection	See the fbInvalidConnection entry in the <b>FIRST CALL DEPTH</b> column previously in this table.
		frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fGetWorkerRecord	ClearWorkerUDT	See the ClearWorkerUDT entry	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	ftGetUpdateFlag	See the ftGetUpdateFlag entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.SetVisibleProperty		
	swss.BreakUpLoadNumber		
	swss.IncrementMessageDots		
	chkLOA_Coord_Click	See the chkLOA_Coord_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	swss.SetEnabledProperty		
	fGetWorkerBasics	See the fGetWorkerBasics entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fGetWorkerPhone	See the fGetWorkerPhone entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fGetWorkerAddress	See the fGetWorkerAddress entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fGetWorkerLanguages	See the fGetWorkerLanguages entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fGetWorkerLoadPrograms	See the fGetWorkerLoadPrograms entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fGetWorkerCoDist	See the fGetWorkerCoDist entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fGetWorkerADPT_SpvSrCo	See the fGetWorkerADPT_SpvSrCo entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fGetWorkerRoles	swss.FillQueriedResultset	
		{SWSS.IDENTIFY_COORD.RETURN_ROLES}	
		If the {SWSS.	See the

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		IDENTIFY_COORD} package is invalid, then Form_QueryUnload.	Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
		If an error occurs, then →	frmError.Form_Load
			mnuFileExit_Click
	fGetWorkerHours	See the fGetWorkerHours entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fEnableOrDisableAllData Fields	See the fEnableOrDisableAllDataFields entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fCoordinatorCountiesEnabled	See the fCoordinatorCountiesEnabled entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fADPT_SpvSrCountiesEnabled	See the fADPT_SpvSrCountiesEnabled entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fEnableFieldLabelPairs	swss.SetEnabledProperty	
		If an error occurs, then →	frmError.Form_Load
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fInitializeAddMode	cboApplication_Click	See the cboApplication_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	swss.SetVisibleProperty		
	fEnableOrDisableAllData Fields	See the fEnableOrDisableAllDataFields entry in the <b>FIRST CALL</b>	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		<b>DEPTH</b> column previously in this table.	
	cboSecurityLevel_Click	See the cboSecurityLevel_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	fClearMaskedTextBox	See the fClearMaskedTextBox entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	swss.SetEnabledProperty		
	chkAccess(0-7)_Click	See the chkAccess_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	cboLanguages(0-2)_Click	See the cboLanguages_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	chkLOA_Coord_Click	See the chkLOA_Coord_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	fCoordinatorCountiesEnabled	See the fCoordinatorCountiesEnabled entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fDetermineApplicationMode	See the fDetermineApplicationMode entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	ClearWorkerUDT	See the ClearWorkerUDT entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	cboCountyCodeListBox_Click	See the cboCountyCodeListBox_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	fGetCountyDependentWo	See the	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	rkrs	fGetCountyDependentWorkers entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.
fInititalizeEditMode	cboApplication_Click	See the cboApplication_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.	
	fDetermineApplicationMo de	See the fDetermineApplicationMode entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	swss.SetVisibleProperty		
	chkAccess(0-5)_Click	See the chkAccess_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.	
	millerm7.tBuildSQLStrin g		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	fGetAllWorkers	See the fGetAllWorkers entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	cboLoadNumberList_Clic k	See the cboLoadNumberList_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE:</b> ADDSTAFF.FRM (Control Events)” table previously in this document.	
	ClearWorkerUDT	See the ClearWorkerUDT entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	fClearDataFields	See the fClearDataFields entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	swss.SetEnabledProperty		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fMenuItemsEnabled	swss.SetEnabledProperty		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.
fOneTimeInitialization	fGetSecurityLevels	See the fGetSecurityLevels entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fGetLanguages	See the fGetLanguages entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	millerm7.SendAppNameToDB		
	cboCountyCodeListBox_Click	See the cboCountyCodeListBox_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	cboApplication_Click	See the cboApplication_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	swss.SetEnabledProperty		
	If an error occurs, then	frmError.Form_Load	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	→		
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fReconciliation	millerm7.tBuildSQLString		
	clsConn.Execute		
	{SWSS.reconpkg.reconcile_info}		
	millerm7.TicklerShells		
	mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.	
	If an error occurs, then →	frmError.Form_Load	
fSelectedDistrict	fWorkerAccessNotice	See the fWorkerAccessNotice entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM (Control Events)</b> ” table previously in this document.
fSetActionPanelVisible	swss.SetVisibleProperty		
	swss.bSaferSetFocus		
	If an error occurs, then →	frmError.Form_Load	
fSetDataPanelCaptions	If an error occurs, then →	frmError.Form_Load	
fSetDefaultSecurityLevel	cboSecurityLevel_LostFocus	See the cboSecurityLevel_LostFocus entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b>	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		(Control Events)" table previously in this document.	
	cboSecurityLevel_Click	See the cboSecurityLevel_Click entry in the <b>FIRST CALL DEPTH</b> column of the "MODULE: ADDSTAFF.FRM (Control Events)" table previously in this document.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the "MODULE: ADDSTAFF.FRM (Control Events)" table previously in this document.
fSetHeaderLabel	swss.SetEnabledProperty		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the "MODULE: ADDSTAFF.FRM (Control Events)" table previously in this document.
fSetWorkerPanelA_Visible	swss.SetVisibleProperty		
	swss.bSaferSetFocus		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the "MODULE: ADDSTAFF.FRM (Control Events)" table previously in this document.
fSetWorkerPanelB_Visible	swss.SetVisibleProperty		
	swss.bSaferSetFocus		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the



July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
			mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the <b>"MODULE: ADDSTAFF.FRM (Control Events)"</b> table previously in this document.
fSetWorkerPanelC_Visible	swss.SetVisibleProperty		
	swss.bSaferSetFocus		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the <b>"MODULE: ADDSTAFF.FRM (Control Events)"</b> table previously in this document.
fSynchSWSS_CaseTable	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Procedure		
	{SWSS.WORKER_MOD.s ync_worker_ids}		
	If the {SWSS.WORKER_MOD} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the <b>"MODULE: ADDSTAFF.FRM (Control Events)"</b> table previously in this document.	
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the <b>"MODULE: ADDSTAFF.FRM (Control Events)"</b> table previously in this document.
ftExistingOracleAccount	millerm7.tBuildSQLString		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	clsConn.Execute		
	{SWSS.Utilities.oracle_ac count_exists}		
	If an error occurs, then →	frmError.Form_Load	
		swss.SetEnabledProperty	
ftGetUpdateFlag	millerm7.tBuildSQLStrin g		
	tGenericMessage	See the tGenericMessage entry in the <b>FIRST CALL DEPTH</b> column later in this table.	
	millerm7.CallSQL_Proce dure		
	{SWSS.Utilities.load_no_ update_flag }		
	If the {SWSS.Utilities} package is invalid, then Form_QueryUnload.	See the Form_QueryUnload entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	If an error occurs, then →	frmError.Form_Load	
fUserFieldAccess	fEnableFieldLabelPairs	swss.SetEnabledProperty	
		If an error occurs, then →	frmError.Form_Load
	ftGetUpdateFlag	See the ftGetUpdateFlag entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	fSetDefaultSecurityLevel	See the fSetDefaultSecurityLevel entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	chkAccess(0-4)_Click	See the chkAccess_Click entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	swss.SetEnabledProperty		
	cboAlternateList_KeyDow n	See the cboAlternateList_KeyDown entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b> (Control Events)” table previously in this document.	
	cboSupervisorList_KeyDo wn	See the cboSupervisorList_KeyDown entry in the <b>FIRST CALL DEPTH</b> column of the “ <b>MODULE: ADDSTAFF.FRM</b>	

July 06, 2001

<b>MODULE: ADDSTAFF.FRM</b> (Procedures)			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		(Control Events)" table previously in this document.	
	fEnabledFrameFields	See the fEnabledFrameFields entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	If an error occurs, then →	frmError.Form_Load	
fWorkerAccessNotice	chkAccess(0-4)_Click	See the chkAccess_Click entry in the <b>FIRST CALL DEPTH</b> column of the " <b>MODULE: ADDSTAFF.FRM</b> (Control Events)" table previously in this document.	
	swss.ReplaceString		
	swss.SetVisibleProperty		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	See the mnuFileExit_Click entry in the <b>FIRST CALL DEPTH</b> column of the " <b>MODULE: ADDSTAFF.FRM</b> (Control Events)" table previously in this document.
SSN_VALIDATE	If an error occurs, then →	frmError.Form_Load	
tGenericMessage	If an error occurs, then →	frmError.Form_Load	

<b>MODULE: InactiveLoadNumbers.FRM</b>			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
cmdChooseActionScreen_Click	Form_QueryUnload		
Form_Load	swss.SetDeviceIndependentWindow		
	fGetInactiveLoadNumbers	millerm7.tBuildSQLString	
		millerm7.CallSQL_Procedure	
		{SWSS.UTILITIES.get_inactive_loads}	
		If the {SWSS.UTILITIES} package is invalid, then Form_QueryUnload	
		swss.tBuildLoadNumber	
		millerm7.gtConvertOracleDate	

July 06, 2001

<b>MODULE: InactiveLoadNumbers.FRM</b>			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		If an error occurs, then →	frmError.Form_Load
			Form_QueryUnload
	swss.SetEnabledProperty		

<b>MODULE: frmPDC_ModCaseSelect.FRM</b>			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
cmdCancelPDC_Mod_Click	mnuFileExit_Click	Form_QueryUnload	
cmdCancelRejectCase_Click	swss.SetVisibleProperty		
	swss.bSaferSetFocus		
cmdOK_LogNumber_Click	millerm7.GetCaseInformation		
	swss.PrivateINI_Register	swss.tGetAppPath	
	swss.lPrivateINI_PutString		
	ftLegalStatusDesc	millerm7.tBuildSQLString	
		swss.CallSQL_Procedure	
		{SWSS.packlists.get_legal_status}	
		If an error occurs, then →	frmError.Form_Load
	swss.tCaseStateDescription		
	ftLivingArrangementDesc	millerm7.tBuildSQLString	
		swss.CallSQL_Procedure	
		{SWSS.packlists.get_liv_arr}	
		If an error occurs, then →	frmError.Form_Load
	ftGoalCodeDesc	millerm7.tBuildSQLString	
		swss.CallSQL_Procedure	
		{SWSS.packlists.get_goals}	
		If an error occurs, then →	frmError.Form_Load
	ftCountyName	millerm7.tBuildSQLString	
		swss.CallSQL_Procedure	
		{SWSS.packlists.get_county}	
		If an error occurs, then →	frmError.Form_Load
	swss.SetVisibleProperty		
	swss.bSaferSetFocus		
	cmdCancelRejectCase_Click	See the cmdCancelRejectCase_Click entry in the <b>FIRST CALL DEPTH</b> column	

July 06, 2001

<b>MODULE: frmPDC_ModCaseSelect.FRM</b>			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		previously in this table.	
	swss.PrivateINI_Register	swss.tGetAppPath	
	swss.bPrivateINI_Section Exists		
	swss.PrivateINI_DeleteSection		
	swss.ExtractINI_Info		
	If an error occurs, then →	frmError.Form_Load	
		mnuFileExit_Click	Form_QueryUnload
cmdOK_SelectCase_Click	swss.CommonShell		
	swss.PrivateINI_Register	swss.tGetAppPath	
	swss.bPrivateINI_Section Exists		
	swss.PrivateINI_DeleteSection		
	mnuFileExit_Click	Form_QueryUnload	
Form_Load	swss.SetDeviceIndependentWindow		
	swss.SetEnabledProperty		
	cmdOK_LogNumber_Click	See the cmdOK_LogNumber_Click entry in the <b>FIRST CALL DEPTH</b> column previously in this table.	
	swss.SetVisibleProperty		

<b>MODULE: frmUtilitiesSplash.FRM</b>			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
Form_KeyPress	Form_QueryUnload		
Form_Load	GetVersionEx	API call from <b>kernel32.dll</b>	
Frame1_Click	Form_QueryUnload		

<b>MODULE: SupervisorReports.FRM</b>			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
cmdChooseActionScreen_Click	Form_QueryUnload		
cmdPrintScreen_Click	mnuFilePrint_Click		
Form_Load	swss.SetDeviceIndependentWindow		
	fGetMultipleLoadList	fCommonQuerySetup	See the fCommonQuerySetup entry in the <b>FIRST CALL DEPTH</b> column later in this table.
		swss.CallSQL_Procedure	
		{SWSS.UTILITIES.get_logi	

July 06, 2001

<b>MODULE: SupervisorReports.FRM</b>			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		n_names}	
		If the {SWSS.UTILITIES} package is Invalid, then Form_QueryUnload	
		{SWSS.UTILITIES.get_wor ker_info}	
		{SWSS.UTILITIES.get_loa d_programs}	
		If an error occurs, then →	frmError.Form_Load
			Form_QueryUnload
	fGetSubordinateLoadList	fCommonQuerySetup	See the fCommonQuerySetup entry in the <b>FIRST CALL DEPTH</b> column later in this table.
		swss.CallSQL_Procedure	
		{SWSS.UTILITIES.GetSup rvsdLoads}	
		If the {SWSS.UTILITIES} package is Invalid, then Form_QueryUnload	
		If an error occurs, then →	frmError.Form_Load
	swss.SetVisibleProperty		Form_QueryUnload
	swss.SetEnabledProperty		

<b>MODULE: Stored SQL Procedures: SWSS.UTILITIES package</b>			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
get_co_dist	SWSS.IDENTIFY_COORD .CENTRAL_SEC	SWSS.FIND_DB_USER	
	SWSS.IDENTIFY_COORD .LOCAL_SEC	SWSS.FIND_DB_USER	
get_security_codes			
get_wrkr_load_no	SWSS.IDENTIFY_COORD .CENTRAL_SEC	SWSS.FIND_DB_USER	
NoOfActiveLoadNos			
get_sup_workers	SWSS.CENTRAL_OFFICE .G_CENTRAL_OFFICE_C ODE	(This is a global constant, not a subroutine)	
get_worker_info	Worker_LoadNo		
	AWorker_Name		
	LoadWorkerName		
get_worker_phone			
get_load_programs	SWSS.CENTRAL_OFFICE .G_CENTRAL_OFFICE_C ODE	(This is a global constant, not a subroutine)	
get_security_co_dist			
get_ADPT_sup_co			
get_worker_hrs			
VERIFY_SUPERVISOR	SUP_NOT_IN_HIERARCH Y		

July 06, 2001

<b>MODULE: Stored SQL Procedures: SWSS.UTILITIES package</b>			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
load_no_update_flag	SWSS.IDENTIFY_COORD .CENTRAL_SEC	SWSS.FIND_DB_USER	
	SWSS.IDENTIFY_COORD .LOCAL_SEC	SWSS.FIND_DB_USER	
	load_assigned2wrkr		
	loc_sec_load_no	SWSS.IDENTIFY_COORD .RETURN_ROLES	
oracle_account_exists	SWSS.IDENTIFY_COORD .RETURN_ROLES		
	SWSS.IDENTIFY_COORD .CENTRAL_SEC	SWSS.FIND_DB_USER	
	SWSS.CENTRAL_SECURITY_ADMIN.GRANT_REVOKE_ROLE		
	SWSS.IDENTIFY_COORD .LOCAL_SEC	SWSS.FIND_DB_USER	
	SWSS.LOCAL_SECURITY_ADMIN.GRANT_REVOK E_ROLE		
CheckForLoginName			
get_login_names			
duplicate_active_load			
get_active_cases			
get_county_address			
get_district_address			
get_inactive_loads	SWSS.CENTRAL_OFFICE .G_CENTRAL_OFFICE_C ODE	(This is a global constant, not a subroutine)	
	SWSS.IDENTIFY_COORD .CENTRAL_SEC	SWSS.FIND_DB_USER	
get_worker_address			
get_worker_languages			
OtherAssgndProgs			
get_alt_workers			
GetSuprvsdLoads			
IsSpecialAddress			
get_county_districts	(no longer used)		
get_worker_load_no	(no longer used)		
get_all_districts	(no longer used)		
reconcile_case	(no longer used)		
(function)	(no longer used)		
Get_Statewide_Flag			

<b>MODULE: Stored SQL Procedures: SWSS.WORKER_MOD package</b>			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
save_LOA2_worker			
save_worker_load			
CheckLoadNoMatch			
save_new_profile	LoadNmbrConflict		

July 06, 2001

<b>MODULE: Stored SQL Procedures: SWSS.WORKER_MOD package</b>			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
	SWSS.IDENTIFY_COORD .CENTRAL_SEC	SWSS.FIND_DB_USER	
	SWSS.CENTRAL_SECURITY_ADMIN.CREATE_LCL _SEC_COORD	SWSS.IDENTIFY_COORD .CENTRAL_SEC	SWSS.FIND_DB_USER
	SWSS.LOCAL_SECURITY _ADMIN.CREATE_SWSS_ WORKER	SWSS.IDENTIFY_COORD .LOCAL_SEC	SWSS.FIND_DB_USER
		SWSS.IDENTIFY_COORD .CENTRAL_SEC	SWSS.FIND_DB_USER
	SWSS.IDENTIFY_COORD .LOCAL_SEC	SWSS.FIND_DB_USER	
	AdminWorkerTable		
	AdminSecurityTable		
	AdminLoadTable	AdmnLdWrkrTbl	
	worker_ids_synced		
	AdminLoadProgram		
	SWSS.SWSS_SET_DEF_S CHEMA_PKGS	(See database team documentation for information on this stored procedure.)	
save_worker_phone			
save_ADPT_sup_co			
delete_ADPT_sup_co			
save_worker_languages			
save_worker_address			
save_security_assign	coord_rec_not_exist		
save_worker_name			
save_security_code			
save_load_program			
save_worker_hrs			
sync_worker_ids			
inactivate_load_no	SWSS.IDENTIFY_COORD .RETURN_ROLES		
	SWSS.IDENTIFY_COORD .CENTRAL_SEC	SWSS.FIND_DB_USER	
	SWSS.CENTRAL_SECURITY_ADMIN.GRANT_REVO KE_ROLE		
	SWSS.IDENTIFY_COORD .LOCAL_SEC	SWSS.FIND_DB_USER	
	SWSS.LOCAL_SECURITY _ADMIN.GRANT_REVOK E_ROLE		
ConvertLOA_Load	SWSS.IDENTIFY_COORD .CENTRAL_SEC	SWSS.FIND_DB_USER	
	SWSS.CENTRAL_SECURITY_ADMIN.CREATE_LCL _SEC_COORD	SWSS.IDENTIFY_COORD .CENTRAL_SEC	SWSS.FIND_DB_USER
	SWSS.LOCAL_SECURITY _ADMIN.CREATE_SWSS_ WORKER	SWSS.IDENTIFY_COORD .LOCAL_SEC	SWSS.FIND_DB_USER
		SWSS.IDENTIFY_COORD	SWSS.FIND_DB_USER



July 06, 2001

<b>MODULE: Stored SQL Procedures: SWSS.WORKER_MOD package</b>			
<b>FIRST CALL DEPTH</b>	<b>SECOND CALL DEPTH</b>	<b>THIRD CALL DEPTH</b>	<b>FOURTH CALL DEPTH</b>
		.CENTRAL_SEC	
	SWSS.IDENTIFY_COORD .LOCAL_SEC	SWSS.FIND_DB_USER	
	AdminSecurityTable		
	AdminLoadTable	AdmnLdWrkrTbl	
	AdminLoadProgram		
	SWSS.SWSS_SET_DEF_S CHEMA_PKGS	(See database team documentation for information on this stored procedure.)	
add_load_to_worker	(no longer used)		
reassign_active_case	(no longer used)		

## Screen Images

Print out the most “all-encompassing” version of the screens, which may have elements or panels that display or change based on user input. Point out any specific navigation quirks or on screen stuff, such as under what conditions various panels display or disappear. Discuss when the buttons say: “Continue” and when they say “Continue Flow” or “Print 5S”.

## Report (output) Images

Print out versions of each output report generated by the module. For each image, explain its usage.

Not Applicable.

## Data Elements

The following table represents all fields for which data is read from the database as well as any fields in which users enter data and store it in the database.

<b>ADDSTAFF.FRM</b>					
<b>Form Field Description</b>	<b>Data Source</b>	<b>Data Target</b>	<b>Target Type</b>	<b>Constraints</b>	<b>Reqd</b>
<b>Screen1:</b>					
SSN (Social Security Number)	swss.worker.ssn	swss.worker.ssn	varchar2(10)		Y
Application	swss.load.project_ind	swss.load.project_ind	varchar2(1)	S, L, B	Y
Worker	swss.worker.worker_name swss.load.load_no swss.load.project_ind (when = B)				
Worker Security Level	swss.security.security_code	swss.security.security_code	number(2)	value must be in swss.security_codes .security_code	Y (SWSS) N (LOA2)
	When swss.worker.phone.phone_type_code = 'P1' for corresponding record tied to swss.phone.phone_id				
Phone	swss.phone.area_code swss.phone.phone_no	swss.phone.area_code swss.phone.phone_no	varchar2(5) varchar2(10)		

ADDSTAFF.FRM					
Form Field Description	Data Source	Data Target	Target Type	Constraints	Reqd
Ext (Phone number extension)	swss.worker_phone.ext ension	swss.worker_phone.ext ension	varchar2(5)		
Hours 1 (Phone Availability)	loa.worker_hours.availa bility1	loa.worker_hours.availa bility1	varchar2(100)		
Hours 2 (Phone Availability)	loa.worker_hours.availa bility2	loa.worker_hours.availa bility2	varchar2(100)		
<b>I. Name</b>	<i>Last, Suffix, First, and Middle names are also concatenated together and stored in the worker_name column of the swss.worker table</i>				
Last	swss.worker.last_name	swss.worker.last_name	varchar2(30)		Y
Suffix	swss.worker.name_suffi x	swss.worker.name_suffi x	varchar2(3)		
First	swss.worker.first_name	swss.worker.first_name	varchar2(30)		
Middle	swss.worker.middle_na me	swss.worker.middle_na me	varchar2(30) (only one character used)		
<u>Username</u>	swss.security.login_nam e	swss.security.login_nam e	varchar2(30)	user may only enter the underscore character (" _"), digits 0-9 and letters A-Z (small and caps) or user may use the Backspace key. All other keystrokes are invalid	Y (SWSS) N (LOA2)
<b>II. Load Number Information</b>	<i>all fields are also concatenated together to form an entry in the load_no column of the swss.load table</i>				
County	swss.load.county_no	swss.load.county_no	number(2)	00 < n <= 84, value must be in swss.county.county_no	Y
District	swss.load.district_no	swss.load.district_no	number(2)	value must be in swss.district.district_no	Y
Section	swss.load.section_no	swss.load.section_no	number(2)		Y
Unit	swss.load.unit_no	swss.load.unit_no	number(2)		Y
Worker	swss.load.worker_no	swss.load.worker_no	number(2)	value must be 77 when security level = 33, otherwise, value may be any number from 00 to 99, <i>except</i> 77.	Y
LOA2 Worker	swss.load.project_ind	swss.load.project_ind	varchar2(1)	value of swss.load.project_ind = 'B' when form field is checked.	
<b>III. Password</b>	<i>only pertains to the password for an Oracle account</i>				

**Screen2:**

**I. Worker Update Access**

July 06, 2001

ADDSTAFF.FRM					
Form Field Description	Data Source	Data Target	Target Type	Constraints	Reqd
Protective Services (CPS)	swss.load_program.program_code	swss.load_program.program_code	varchar2(4)	Not Null, value must be in swss.program.program_code	
Child Foster Care (CFC)	swss.load_program.program_code	swss.load_program.program_code	varchar2(4)	Not Null, value must be in swss.program.program_code	
Adoption (ADPT)	swss.load_program.program_code	swss.load_program.program_code	varchar2(4)	Not Null, value must be in swss.program.program_code	
Juvenile Justice (JJ)	swss.load_program.program_code	swss.load_program.program_code	varchar2(4)	Not Null, value must be in swss.program.program_code	
Provider Management	swss.load.provider_mgmt_ind	swss.load.provider_mgmt_ind	varchar2(1)	Y, N	
Security Administration	<i>enables Screen 4 and assigns SWSS_FIELD_ADMIN Oracle role to Username...when checked</i>				

## II. Worker Relationships

Alternate	swss.worker.worker_name	swss.load.alt_worker_id	number		
Supervisor	swss.load.load_no swss.worker.worker_name swss.load_worker.supv_load_no	swss.load_worker.supv_load_no	number number(10)	security code of load_no must be <= 30	Y if security code >= 60, otherwise N

## III. Adoption Supervisor County Assignments

Counties Available	swss.county.county_no				
Counties Assigned	swss.co_adoption_supv_sr.county_no	swss.co_adoption_supv_sr.county_no	number(2)		Y if this is a Security Administration profile.

## Screen 3: I. Address

	<i>When swss.worker_address.address_type_code = 'P' for corresponding record tied to swss.address.address_id or swss.county_address.address_id</i>				
Primary: Line 1	swss.address.line1	swss.address.line1	varchar2(60)		
Primary: Line 2	swss.address.line2	swss.address.line2	varchar2(60)		
Primary: City	swss.address.city	swss.address.city	varchar2(30)		
Primary: State	constant	swss.address.state_code	varchar2(2)	Always value of 'MI'.	
Primary: Zipcode	swss.address.zip	swss.address.zip	varchar2(10)		
Primary: Zip + 4	swss.address.zip_plus	swss.address.zip_plus	varchar2(10)		
Primary: E-Mail	swss.worker.email_address	swss.worker.email_address	varchar2(100)		
	<i>When swss.worker_address.address_type_code = 'M' for</i>				

ADDSTAFF.FRM					
Form Field Description	Data Source	Data Target	Target Type	Constraints	Reqd

	<i>corresponding record tied to swss.address.address_id or swss.county_address.address_id</i>				
Mailing: Line 1	swss.address.line1	swss.address.line1	varchar2(60)	Always value of 'MI'	
Mailing: Line 2	swss.address.line2	swss.address.line2	varchar2(60)		
Mailing: City	swss.address.city	swss.address.city	varchar2(30)		
Mailing: State	Constant	swss.address.state_code	varchar2(2)		
Mailing: Zipcode	swss.address.zip	swss.address.zip	varchar2(10)		
Mailing: Zip + 4	swss.address.zip_plus	swss.address.zip_plus	varchar2(10)		

## II. Phones

	<i>When swss.worker_phone.phone_type_code = 'F' for corresponding record tied to swss.phone.phone_id</i>				
Fax	swss.phone.area_code	swss.phone.area_code	varchar2(5)		
	swss.phone.phone_no	swss.phone.phone_no	varchar2(10)		
Ext	swss.worker_phone.extension	swss.worker_phone.extension	varchar2(5)		

## III. Languages

swss.worker_language.language_code	swss.worker_language.language_code	varchar2(2)	Not Null, value must be in swss.language.language_code
swss.language.description			

## Screen 4:

### I. Security Coordinator County-District Assignments

List of Counties	swss.county.county_no swss.county.county_name				
List of County Districts	swss.district.district_no swss.district.district_name				
County-District Assignments	swss.co_security_coord.county_no swss.county.county_name swss.co_security_coord.district_no swss.district.district_name	swss.co_security_coord.county_no swss.co_security_coord.district_no	number(2) number(2)	Not Null Not Null	

### Data Elements not displayed:

	SYS.DBA_ROLE_PRIVS.GRANTED_ROLE				
gtPrograms	swss.load_program.program_code				
gtLogNumber	swss.swss_case.log_id				Y
gtCaseState	swss.swss_case.swss_case_state				
gtCaseCounty	swss.swss_case.county_no				

July 06, 2001

ADDSTAFF.FRM					
Form Field Description	Data Source	Data Target	Target Type	Constraints	Reqd

When Inactivate Mode is checking for active cases assigned to the inactivated load number.

swss.load\_case.load\_no  
swss.swss\_case.case\_n  
ame

InactiveLoadNumbers.FRM					
Form Field Description	Data Source	Data Target	Target Type	Constraints	Reqd
Load Number	swss.load_worker.load_no				
Worker Name	swss.worker.worker_name				
Username	swss.load_worker.login_name				
Start Date	swss.load_worker.begin_date				
End Date	swss.load_worker.end_date				

frmPDC_ModCaseSelect.FRM					
Form Field Description	Data Source	Data Target	Target Type	Constraints	Reqd
<b>I. Enter Case's Log Number Panel</b>					
Log Number	swss.swss_case.log_id				
<b>II. Case Summary Panel</b>					
Log Number	swss.swss_case.log_id				
Case Name	swss.group_person.last_name, swss.group_person.first_name				
Case Number	swss.swss_case.cis_case_no				
Legal Status Code	swss.swss_case.latest_legal_status swss.legal_status_code.legal_status_text				
Case County	swss.swss_case.county_no or swss.swss_case.transfer_to_county swss.county.county_name				
Case Program Code	swss.swss_case.program_code				
Latest Living Arrangement	swss.swss_case.latest_living_arrangement swss.living_arrangement.description				
Goal Code	swss.cfc_info.goal_code swss.goal.description				
Case Status/State	swss.swss_case.swss_case_state				
Confidentiality Indicator	swss.swss_case.high_profile_indicator				

SupervisorReports.FRM					
Form Field Description	Data Source	Data Target	Target Type	Constraints	Reqd
<b>I. Multiple Load Number Report</b>					

SupervisorReports.FRM					
Form Field Description	Data Source	Data Target	Target Type	Constraints	Reqd
<Username>* Load Number	swss.load_worker_opt1.load_no				
Load Number's Program Access	swss.load.provider_mgt_ind swss.load_program.program_code				
Name of Load Number's Supervisor	swss.worker.worker_name				
Supervisor's Load Number	swss.load_worker_opt1.supv_load_no				
<b>II. Subordinate List</b>					
<Username>* Load Number	swss.load_worker_opt1.supv_load_no				
Subordinate's Name	swss.worker.worker_name				
Subordinate's Load Number	swss.load_worker_opt1.load_no				
Subordinate's Begin Date	swss.load_worker_opt1.begin_date				
*<Username> is a variable that refers to the username of the profile that the current user has selected.					

List of database tables (unless specified, assume table is in the SWSS schema)

- worker
- load
- security
- worker\_phone
- phone
- security\_codes (read-only)
- loa.worker\_hours
- county (read-only)
- district (read-only)
- load\_program
- program (read-only)
- co\_adoption\_supvsr
- worker\_address
- address
- county\_address (read-only)
- district\_address (read-only)
- worker\_language
- language (read-only)
- co\_security\_coord
- SYS.DBA\_ROLE\_PRIVS (read-only ?)
- swss\_case
- load\_case
- load\_worker
- load\_worker\_opt1 (read-only)
- all\_users (read-only)
- swss\_conversion (read-only)
- worker\_name\_hist (write-only)
- V\$SESSION (read-only ?)
- dba\_users (read-only)
- SYS.DBA\_ROLES (read-only)
- From the swss\_set\_def\_schema\_pkgs stored procedure
  - See database team documentation

## Integration with Existing System

Access to this module is available as a selection from the Common Menu Bar in all modules as well as an icon found in the Main Menu module. Also, if the user is either a Security Coordinator or a PDC User, the user will be taken directly to the Utilities module immediately after logging into the system at the SWSS Login module. Access to any other SWSS module (except PDC Users who can go to the Payment Document Control Utility) for these special users is denied.

Some of the functions available in this module are accessible only by certain users. The details of who is allowed access to perform what “action” is specified in the Utilities User Requirements document. However, the method that Utilities uses to identify the current user involves reading several entries in the SWSS\_INI.INI file. These entries are:

- [Current User]
  - Program(s)
    - Helps determine grant/denial of access to all actions except the Change Staff Profile action on the Choose Action screen.
  - Security Level
    - Helps determine grant/denial of access to the Payment Document Control Utility action on the Choose Action screen.
- [Other]
  - Central Office Code
    - Helps determine grant/denial of access to the PDC User security level, the Adoption Supervisor County Assignments frame on Screen 2, and the Security Administration checkbox on Screen 2.
- [Case Info]
  - Log Number
    - Helps determine grant/denial of access to the CPS Transfer Information action on the Choose Action screen.
  - Case Status
    - Helps determine grant/denial of access to the CPS Transfer Information action on the Choose Action screen.

When a user changes his/her own password, the Password entry of the [Current User] section in the SWSS\_INI.INI file must be changed to reflect a password change in the Oracle database for the current user. If the encrypted password stored in this entry of the INI file stops matching the database password for the current user, the user will be prevented from accessing any other SWSS modules when s/he leaves the Utilities module. It will cause one or more error(s) and force the user to terminate his/her SWSS session and log back into the system.

## Module Dependencies

If this module depends on other modules, or if other modules depend on this module, to an extent beyond what has been described in the “Integration with Existing System” section above, such as Medicaid requiring both Placement and Funding data, list and explain those dependencies here.

The very issue of access to the SWSS application is determined and maintained in the Utilities module for all SWSS users except the Central Office Security Coordinators (the ones with the SWSS\_CENTRAL\_ADMIN role). All data for the staff profiles entered in this module has an effect on each profile in all other modules. This module determines if a user:

- can add, modify, and/or inactivate other staff profiles
- is a supervisor, a PDC User, a regular worker, or a clerical worker
- is an alternate worker to another profile

- is a supervisor of another profile
- is a County Adoption Supervisor and for which county or counties
- can be assigned as a primary worker for Child Protective Services (CPS doesn't really do anything in the current SWSS FAJ implementation), Child Foster Care, Adoption, and/or Juvenile Justice cases
- is a manager of Provider data

This module also:

- sets up load numbers as LOA2, SWSS, or LOA2/SWSS
- stores worker address and phone number information which is printed off on form letters
- provides a link for SWSS workers between the SWSS FAJ, CIS, ASSIST (and perhaps other) applications by requiring that each profile store a Social Security Number. This is the only way that the Central Office Security Office can track a worker's activities from one FIA applicatin to the next.
- Sets up the profile data that is tested by Common Code to help determine if a user is granted/denied access to SWSS case data and/or SWSS modules.

## Database Subject Area

- Show the tables with fields read, created, or modified here. (CRUD matrix, if you want to be fancy.)
- List the STORED PROCEDURES with detailed descriptions of how that stored procedure works (we're serious. We really need it!)
- Use the Interface specifications that were generated (by Ed?) for the stored procedure calls that the VB module makes.

The following are the stored SQL procedure interface definitions for interacting with the SWSS.UTILITIES package:

```

/*****
*      Get list of all security levels from the security_code table that have a description stored in the *
*      description field. The list is returned by alphabetic order of its description column value.      *
*      This procedure will not return the SuperUser security level.                               *
*****/

```

TYPE Security\_Code\_Rec IS RECORD

```

(
  t_security_code  security_codes.security_code%TYPE,
  t_security_desc  security_codes.description%TYPE
);

```

TYPE Security\_Codes\_Cursor IS REF CURSOR RETURN Security\_Code\_Rec;

PROCEDURE get\_security\_codes

(GetSecurityCodesCursor IN OUT UTILITIES.Security\_Codes\_Cursor);

```

/*****
*      Get address information for specified county. Also, this procedure will only return the *
*      type of address specified in the i_address_type input parameter.                      *
*****/

```

TYPE Cnty\_Addr\_Info\_Rec IS RECORD

```

(
  t_line1      address.line1%TYPE,
  t_line2      address.line2%TYPE,
  t_city       address.city%TYPE,
  t_state_code address.state_code%TYPE,
  t_zip        address.zip%TYPE,
  t_zip_plus   address.zip_plus%TYPE,
  t_addr_id    county_address.address_id%TYPE,

```



July 06, 2001

```
t_co_name      county.county_name%TYPE
);
```

```
TYPE Cnty_Addr_Info_Cursor IS REF CURSOR RETURN Cnty_Addr_Info_Rec;
```

```
PROCEDURE get_county_address
```

```
(i_county_number    IN county_address.county_no%TYPE,
 i_address_type     IN county_address.addr_type_code%TYPE,
 GetCntyAddrInfoCursor IN OUT UTILITIES.Cnty_Addr_Info_Cursor
);
```

```
/*
 *      Get address information for specified district in a particular county.  This
 *      procedure will only return information that applies to the type of address
 *      specified in the address_type input parameter.
 */
```

```
TYPE Dist_Addr_Info_Rec IS RECORD
```

```
(
 t_line1      address.line1%TYPE,
 t_line2      address.line2%TYPE,
 t_city       address.city%TYPE,
 t_state_code address.state_code%TYPE,
 t_zip        address.zip%TYPE,
 t_zip_plus   address.zip_plus%TYPE,
 t_addr_id    district_address.address_id%TYPE,
 t_co_name    county.county_name%TYPE,
 t_dist_name  district.district_name%TYPE
);
```

```
TYPE Dist_Addr_Info_Cursor IS REF CURSOR RETURN Dist_Addr_Info_Rec;
```

```
PROCEDURE get_district_address
```

```
(i_county_number    IN district_address.county_no%TYPE,
 i_district_number  IN district_address.district_no%TYPE,
 i_address_type     IN district_address.addr_type_code%TYPE,
 GetDistAddrInfoCursor IN OUT UTILITIES.Dist_Addr_Info_Cursor
);
```

```
/*
 *      GET LIST OF DISTRICTS ASSOCIATED WITH SELECTED COUNTY
 *      This procedure is no longer called from the Utilities VB program.
 *      However, t_district_no and t_district_name are used by the
 *      get_all_districts procedure.
 */
```

```
TYPE t_district_no    IS TABLE OF district.district_no%TYPE INDEX BY BINARY_INTEGER;
TYPE t_district_name  IS TABLE OF district.district_name%TYPE INDEX BY BINARY_INTEGER;
```

```
/*
 *      RETRIEVE INFORMATION ABOUT A PARTICULAR WORKER
 */
```

```
-- using almost the exact same output variables (with the exception of the
-- alternate workers load number which was previously broken down)
```

```
type Worker_Info_Rec is record (
 o_spvr_ind      security.security_code%TYPE,
 o_provider_mgt_ind load.provider_mgt_ind%TYPE,
 o_alt_load      load.load_no%TYPE,
```

July 06, 2001

```
o_alt_name      worker.worker_name%TYPE,
o_spvr_co       load.county_no%TYPE,
o_spvr_dist     load.district_no%TYPE,
o_spvr_sect     load.section_no%TYPE,
o_spvr_unit     load.unit_no%TYPE,
o_spvr_work     load.worker_no%TYPE,
o_spvr_name     worker.worker_name%TYPE,
o_name_last     worker.last_name%TYPE,
o_name_suffix   worker.name_suffix%TYPE,
o_name_first    worker.first_name%TYPE,
o_name_middle   worker.middle_name%TYPE,
o_worker_id     worker.worker_id%TYPE,
o_login_name    load_worker.login_name%TYPE,
o_ssn          worker.ssn%TYPE
);

type WorkerInfoCursor IS REF CURSOR RETURN Worker_Info_Rec;

PROCEDURE Get_Worker_Info  (
    i_worker_load_no  in    load.load_no%TYPE,
    i_county_number    in out  load.county_no%TYPE,
    Worker_Info_Cursor  IN OUT UTILITIES.WorkerInfoCursor);

/*****
*       The following 2 functions are used to extract the name of the worker based
*       on the worker id AND to extract the load number of the alternate worker
*****/
FUNCTION Worker_LoadNo (i_worker_id IN worker.worker_id%TYPE,
                       i_county_number in county.county_no%type)
    return number;

PRAGMA      restrict_references (Worker_LoadNo,wnds,wnps,rnps);

FUNCTION AWorker_Name (i_worker_id    in    worker.worker_id%type)
    return varchar2;

PRAGMA      restrict_references (AWorker_Name,wnds,wnps,rnps);

FUNCTION LoadWorkerName (i_load_no IN load_worker.load_no%TYPE)
    RETURN VARCHAR2;

PRAGMA      restrict_references (LoadWorkerName,wnds,wnps,rnps);

/*****
*       Gets requested phone record type for specified worker.
*       The format of the phone number output has now been changed so that a space
*       has been inserted between the closing parens and the phone number.
*****/
TYPE Worker_Phone_Rec IS RECORD
(
t_phone_number  VARCHAR2(15),
t_phone_ext     worker_phone.extension%TYPE
);

TYPE Worker_Phone_Cursor IS REF CURSOR RETURN Worker_Phone_Rec;

PROCEDURE get_worker_phone
(i_worker_id    IN worker_phone.worker_id%TYPE,
 i_phone_type   IN worker_phone.phone_type_code%TYPE,
```

July 06, 2001

GetWorkerPhoneCursor IN OUT UTILITIES.Worker\_Phone\_Cursor);

```
/*
*****
*      RETRIEVE ADDRESS RECORD FOR A PARTICULAR WORKER.
*****
*/
```

PROCEDURE get\_worker\_address

```
(i_worker_load_no  IN  load_worker.load_no%TYPE,
 i_address_type    IN  worker_address.addr_type_code%TYPE,
 o_line1           OUT address.line1%TYPE,
 o_line2           OUT address.line2%TYPE,
 o_city            OUT address.city%TYPE,
 o_state_code      OUT address.state_code%TYPE,
 o_zip             OUT address.zip%TYPE,
 o_zip_plus        OUT address.zip_plus%TYPE,
 o_address_id      OUT address.address_id%TYPE,
 o_email           OUT worker.email_address%TYPE
);
```

TYPE t\_update\_flag IS TABLE OF VARCHAR2(1) INDEX BY BINARY\_INTEGER;

```
/*
*****
*      This procedure determines the update flag for the passed load number and returns that *
*      flag. Also fixed the problem of allowing one local security coordinator the ability to *
*      change the records of another security coordinator.
*      The possible values of the update flag are:
*      U = Update all fields
*      P = May change password only
*      R = May view records but may not change any fields
*****
*/
```

PROCEDURE load\_no\_update\_flag

```
(i_worker_id      IN  worker.worker_id%TYPE, --The worker_id of the current user.
 i_load_no        IN  load.load_no%TYPE,    --The load number currently "selected".
 o_update_flag    OUT  VARCHAR2);
```

TYPE t\_language\_code IS TABLE OF worker\_language.language\_code%TYPE INDEX BY BINARY\_INTEGER;

```
/*
*****
*      Gathers list of languages currently assigned to worker.
*****
*/
```

TYPE Worker\_Lang\_Rec IS RECORD

```
(
 t_language_code worker_language.language_code%TYPE
);
```

TYPE Worker\_Lang\_Cursor IS REF CURSOR RETURN Worker\_Lang\_Rec;

PROCEDURE get\_worker\_languages

```
(i_worker_id      IN  worker_language.worker_id%TYPE,
 GetWorkerLangCursor IN OUT UTILITIES.Worker_Lang_Cursor
);
```

TYPE t\_worker\_load IS TABLE OF load\_worker.load\_no%TYPE INDEX BY BINARY\_INTEGER;

```
/*
*****
*      Get list of active load numbers with usernames for a particular worker id.
*****
*/
```

TYPE Login\_Name\_Rec IS RECORD

July 06, 2001

---

```
(  
t_user_name load_worker.login_name%TYPE,  
t_worker_load load_worker.load_no%TYPE  
);
```

```
TYPE Login_Names_Cursor IS REF CURSOR RETURN Login_Name_Rec;
```

```
PROCEDURE get_login_names  
(i_worker_id IN worker.worker_id%TYPE,  
 GetLoginNamesCursor IN OUT UTILITIES.Login_Names_Cursor  
);
```

```
TYPE t_log_id IS TABLE OF load_case.log_id%TYPE INDEX BY BINARY_INTEGER;  
TYPE t_case_name IS TABLE OF swss_case.case_name%TYPE INDEX BY BINARY_INTEGER;  
/*****
```

```
* Gets list of active cases assigned to a particular load number. *  
*****/
```

```
TYPE Case_List_Rec IS RECORD
```

```
(  
t_log_id load_case.log_id%TYPE,  
t_case_name swss_case.case_name%TYPE  
);
```

```
TYPE Case_List_Cursor IS REF CURSOR RETURN Case_List_Rec;
```

```
PROCEDURE get_active_cases  
(i_worker_load IN load_case.load_no%TYPE,  
 --o_log_id OUT t_log_id,  
 --o_case_name OUT t_case_name  
 CaseListCursor IN OUT UTILITIES.Case_List_Cursor  
);
```

```
/*****  
* This procedure gets a list of all active county-district combinations that belong *  
* to the passed worker_id. If no records are found, then this indicates that *  
* worker is not a security coordinator. If the only record found returns and *  
* Central Office county and district 00 combination, then the worker_id is a *  
* Central Office Security coordinator with statewide access. *  
*****/
```

```
TYPE Sec_Cnty_District_Rec IS RECORD
```

```
(  
t_county_no district.county_no%TYPE,  
t_district_no district.district_no%TYPE  
);
```

```
TYPE Sec_Cnty_Districts_Cursor IS REF CURSOR RETURN Sec_Cnty_District_Rec;
```

```
PROCEDURE get_security_co_dist  
(i_worker_id IN co_security_coord.security_coord_id%TYPE,  
 GetSecCntyDistrictsCursor IN OUT UTILITIES.Sec_Cnty_Districts_Cursor);
```

```
TYPE t_programs IS TABLE OF VARCHAR2(100) INDEX BY BINARY_INTEGER;  
/*****
```

```
* This procedure was written because we needed a list of program codes assigned to *  
* a particular load number. *  
*****/
```

```
PROCEDURE get_load_programs
```

July 06, 2001

```
(i_load_no IN load_program.load_no%TYPE,  
  o_programs OUT VARCHAR2  
);
```

```
TYPE t_acct_existance IS TABLE OF VARCHAR2(1) INDEX BY BINARY_INTEGER;  
/*****  
* Returns a value, indicating if passed username currently has an Oracle *  
* database account. *  
* The o_account_exists output paramater has four possible values: *  
* Y = The oracle account already exists. Prevent other procedures from *  
* trying to re-create the account. *  
* N = The oracle account does not yet exist. Allow other procedures the *  
* opportunity to create this account. *  
* D = The oracle account already exists, however, the current user does *  
* not have any access rights to adjust the account for access to *  
* SWSS. Any profile modification must stop at this very point. *  
* C = The oracle account exists and the account belongs to a Central *  
* Office Security Coordinator. SWSS should NEVER modify *  
* anything that deals with Central Office Security Coordinator *  
* profiles. All attempts at modifying/creating the current profile *  
* must be stopped until a different username(login_name) is *  
* selected! *  
*****/
```

```
PROCEDURE oracle_account_exists  
(i_login_name IN VARCHAR2,  
  i_assigned_role IN VARCHAR2,  
  o_account_exists OUT VARCHAR2  
);
```

```
/*****  
* Get list of worker names associated with an active load number. *  
*****/
```

```
PROCEDURE duplicate_active_load  
(i_load_no IN load_worker.load_no%TYPE,  
  --o_worker_name OUT t_worker_name  
  o_worker_name OUT VARCHAR2  
);
```

```
/*****  
* This procedure gathers all records in the the co_adoption_supvrsr table that have the *  
* passed load number in the load_no column. *  
*****/
```

```
TYPE ADPT_Sup_Co_Rec IS RECORD  
(  
  t_worker_co load.county_no%TYPE  
);
```

```
TYPE ADPT_Sup_Co_Cursor IS REF CURSOR RETURN ADPT_Sup_Co_Rec;  
PROCEDURE get_ADPT_sup_co -- originally named get_ADPT_supervisor_counties  
(i_load_no IN co_adoption_supvrsr.load_no%TYPE,  
  GetADPT_SupCoCursor IN OUT UTILITIES.ADPT_Sup_Co_Cursor  
);
```

```
/*****  
* Function written by Mark Stock (on 03/10/1999) which will verify that the *  
* passed supervisor worker_id is not in the LOAD table hierarchy below the *  
* passed worker_id. This is an attempt to prevent recursive hierarchies in the *  
*****/
```

July 06, 2001

```
*      LOAD table.      *
***** /
FUNCTION  SUP_NOT_IN_HIERARCHY  -- originally named SUPERVISOR_NOT_IN_HIERARCHY
(
  i_worker_id      IN load_worker.worker_id%TYPE,
  i_supervisor_worker_id IN load_worker.worker_id%TYPE
) RETURN BOOLEAN;

/*****
*      This subroutine was written to call the SUPERVISOR_NOT_IN_HIERARCHY function      *
*      after translating the passed supervisor load number into a worker_id.      *
***** /
PROCEDURE  VERIFY_SUPERVISOR(
  i_worker_id      IN load_worker.worker_id%TYPE,
  i_sup_load_no    IN load_worker.supv_load_no%TYPE,
  o_sup_is_OK      OUT VARCHAR2
);

/*****
*      Created to fulfill LOA2 requirements for SWSS Utilities.      *
***** /
TYPE t_hrs1 IS TABLE OF loa.worker_hours.availability1%TYPE INDEX BY BINARY_INTEGER;
TYPE t_hrs2 IS TABLE OF loa.worker_hours.availability2%TYPE INDEX BY BINARY_INTEGER;
TYPE Worker_Hrs_Rec IS RECORD
(
  t_hrs1 loa.worker_hours.availability1%TYPE,
  t_hrs2 loa.worker_hours.availability2%TYPE
);
TYPE Worker_Hrs_Cursor IS REF CURSOR RETURN Worker_Hrs_Rec;
PROCEDURE get_worker_hrs(
  i_worker_id      IN loa.worker_hours.worker_id%TYPE,
  i_load_no        IN loa.worker_hours.load_no%TYPE,
  o_hours2         OUT t_hrs2*/
  GetWorkerHrsCursor IN OUT UTILITIES.Worker_Hrs_Cursor
);

/*****
*      This procedure gets a list of all active county-district combinations that belong  *
*      to the passed worker_id.      *
*      Commented out the block of code which determines if the passed worker_id has  *
*      a Central Office load number.  As of this date, Central Office users are no longer  *
*      allowed to view/modify/create/inactivate any profiles outside of their  *
*      county-district combinations, even if they are Security Coordinators.  This  *
*      change is documented by the removal of Utilities User Requirements UT-1.8.2.1  *
*      and UT-1.8.3.1.      *
***** /
FUNCTION Get_Statewide_Flag
  ( f_worker_id IN load_worker.worker_id%TYPE
  ) RETURN VARCHAR2;

--
TYPE Cnty_Dist_Rec IS RECORD
(
  --t_county_no    district.county_no%TYPE,
  t_county_no      county.county_no%TYPE,  --Modified this line of code 05/19/2000 - MDM
  t_co_name        county.county_name%TYPE,
  t_district_no    district.district_no%TYPE,
```

July 06, 2001

```
t_district_name district.district_name%TYPE  
);
```

```
TYPE Cnty_Dists_Cursor IS REF CURSOR RETURN Cnty_Dist_Rec;
```

```
PROCEDURE get_co_dist  
(i_worker_id      IN load_worker.worker_id%TYPE,  
  GetCoDistsCursor IN OUT UTILITIES.Cnty_Dists_Cursor  
);
```

```
/*  
* RETRIEVE LIST OF ALL WORKERS IN A PARTICULAR  
* COUNTY WHO ARE SUPERVISORS and who are allowed to supervise  
* passed worker_id. If worker_id is zero, all supervisors in  
* county are returned.  
*/
```

```
TYPE Sup_Worker_Rec IS RECORD
```

```
(  
  t_sup_co      load.county_no%TYPE,  
  t_sup_dist     load.district_no%TYPE,  
  t_sup_sect     load.section_no%TYPE,  
  t_sup_unit     load.unit_no%TYPE,  
  t_sup_worker   load.worker_no%TYPE,  
  t_sup_name     worker.worker_name%TYPE  
);
```

```
TYPE Sup_Worker_Cursor IS REF CURSOR RETURN Sup_Worker_Rec;
```

```
PROCEDURE get_sup_workers -- originally named get_supervisor_workers  
(i_co_number      IN load.county_no%TYPE,  
  i_worker_id     IN worker.worker_id%TYPE,  
  i_proj_ind      IN load.project_ind%TYPE,  
  GetSupWorkersCursor IN OUT UTILITIES.Sup_Worker_Cursor  
);
```

```
/*  
* Checking to see if a database record with the passed i_login_name already  
* exists.  
* This procedure determines if the passed i_address_id refers to  
* a county address or a district address. The i_address_type parameter determines  
* if the procedure is looking for a Primary or a Mailing address.  
* Values of o_IsCoAddr  
* 'T' - Passed i_address_id is indeed a county address.  
* 'F' - Passed i_address_id is not a county address.  
* Values of o_IsDistAddr  
* 'T' - Passed i_address_id is indeed a district address.  
* 'F' - Passed i_address_id is not a district address.  
* If an error occurs, both output parameters will return 'F'.  
*/
```

```
PROCEDURE CheckForLoginName
```

```
(i_login_name IN load_worker.login_name%TYPE,  
  o_worker_id OUT load_worker.worker_id%TYPE,  
  o_worker_name OUT worker.worker_name%TYPE,  
  o_active_flag OUT VARCHAR2);
```

```
/*  
02/01/2000 Matthew D. Miller  
*/
```

July 06, 2001

---

PROCEDURE IsSpecialAddress

```
(i_address_id IN worker_address.address_id%TYPE,  
 i_address_type IN worker_address.addr_type_code%TYPE,  
 o_IsCoAddr OUT VARCHAR2,  
 o_IsDistAddr OUT VARCHAR2  
);
```

```
/*****  
* This procedure determines the number of active load numbers *  
* assigned to the passed worker_id. *  
*****/
```

PROCEDURE NoOfActiveLoadNos

```
(i_worker_id IN load_worker.worker_id%TYPE,  
 o_number_of_load_numbers OUT NUMBER  
);
```

TYPE Wrkr\_Load\_No\_List\_Rec IS RECORD

```
(  
 t_worker_co load.county_no%TYPE,  
 t_worker_dist load.district_no%TYPE,  
 t_worker_sect load.section_no%TYPE,  
 t_worker_unit load.unit_no%TYPE,  
 t_worker_work load.worker_no%TYPE,  
 t_worker_name worker.worker_name%TYPE  
 --t_update_flag VARCHAR2(1)  
);
```

TYPE Wrkr\_Load\_No\_List\_Cursor IS REF CURSOR RETURN Wrkr\_Load\_No\_List\_Rec;

PROCEDURE get\_wrkr\_load\_no

```
(i_worker_id IN worker.worker_id%TYPE,  
 i_proj_ind IN load.project_ind%TYPE,  
 i_vb_executable IN VARCHAR2,  
 i_county_no IN NUMBER,  
 GetWrkrLoadNoListCursor IN OUT UTILITIES.Wrkr_Load_No_List_Cursor);
```

TYPE Inctv\_Load\_No\_List\_Rec IS RECORD

```
(  
 t_worker_co load.county_no%TYPE,  
 t_worker_dist load.district_no%TYPE,  
 t_worker_sect load.section_no%TYPE,  
 t_worker_unit load.unit_no%TYPE,  
 t_worker_work load.worker_no%TYPE,  
 t_worker_name worker.worker_name%TYPE,  
 t_username load_worker.login_name%TYPE,  
 t_date_start load_worker.begin_date%TYPE,  
 t_date_end load_worker.end_date%TYPE  
 --t_update_flag VARCHAR2(1)  
);
```

TYPE Inctv\_Load\_No\_List\_Cursor IS REF CURSOR RETURN Inctv\_Load\_No\_List\_Rec;

PROCEDURE get\_inactive\_loads

```
(i_worker_id IN worker.worker_id%TYPE,  
 i_county_no IN NUMBER,  
 GetInctvLoadNoListCursor IN OUT UTILITIES.Inctv_Load_No_List_Cursor);
```

```
/*****  
* This procedure returns the number of SWSS FAJ programs that the passed *  
* load_no is actively assigned to. *  
*****/
```



July 06, 2001

\*\*\*\*\*  
PROCEDURE OtherAssgndProgs

(i\_worker\_id            IN load\_worker.worker\_id%TYPE,  
  i\_load\_no            IN load\_worker.load\_no%TYPE,  
  o\_other\_assigned\_programs   OUT VARCHAR2);

/\*\*\*\*\*

04/05/2001 Matthew D. Miller

This procedure gathers a list of active load numbers which can be used for assigning an Alternate Worker to a selected staff profile.

If the user is in Add Staff Profile mode (i\_profile\_mode), the list is narrowed down to only active load numbers within the county passed in the i\_county\_no parameter. Local Office Security Coordinator load numbers are filtered out of this list.

If the user is in either Change Staff Profile or Inactivate Staff Profile mode (i\_profile\_mode), this procedure gathers all active load numbers in all counties in which the user assigned to the passed load number, i\_selected\_load\_no, has been assigned an active load number. This procedure does, however, filter out the active load numbers belonging to the user as well as any active load numbers assigned to Local Office Security Coordinators.

Since Local Office Security Coordinators cannot view or change case data, they are useless as Alternate Workers and are therefore filtered out of the list.

The possible values for the i\_profile\_mode input parameter are:

- A - Add Staff Profile Mode
- C - Change Staff Profile Mode
- I - Inactivate Staff Profile Mode

\*\*\*\*\*

TYPE Get\_Alt\_Load\_No\_List\_Rec IS RECORD

(  
  t\_county\_no    load.county\_no%TYPE,  
  t\_district\_no   load.district\_no%TYPE,  
  t\_section\_no    load.section\_no%TYPE,  
  t\_unit\_no       load.unit\_no%TYPE,  
  t\_worker\_no     load.worker\_no%TYPE,  
  t\_project\_ind   load.project\_ind%TYPE,  
  t\_worker\_name   worker.worker\_name%TYPE  
);

TYPE Get\_Alt\_Load\_No\_List\_Cursor IS REF CURSOR RETURN Get\_Alt\_Load\_No\_List\_Rec;

PROCEDURE get\_alt\_workers

( i\_county\_no        IN    load.county\_no%TYPE,  
  i\_selected\_load\_no   IN    load\_worker.load\_no%TYPE,  
  i\_profile\_mode       IN    VARCHAR2,  
  GetAltLoadNoListCursor   IN OUT   UTILITIES.Get\_Alt\_Load\_No\_List\_Cursor  
);

/\*\*\*\*\*

06/04/2001 Matthew D. Miller

This procedure gets a list of load numbers supervised by the i\_worker\_id.

\*\*\*\*\*

TYPE Get\_Supd\_Loads\_List\_Rec IS RECORD

(  
  t\_load\_no        load\_worker.supv\_load\_no%TYPE,  
  t\_sbrdnt\_name    worker.worker\_name%TYPE,  
  t\_sbrdnt\_load    load\_worker.load\_no%TYPE,  
  t\_begin\_date     load\_worker.begin\_date%TYPE  
);

TYPE Get\_Supd\_Loads\_List\_Cursor IS REF CURSOR RETURN Get\_Supd\_Loads\_List\_Rec;

July 06, 2001

---

PROCEDURE GetSuprvsdLoads

```
( i_worker_id      IN    load_worker.worker_id%TYPE,
  GetSupdLoadsListCursor IN OUT UTILITIES.Get_Supd_Loads_List_Cursor
);
```

The following are the stored SQL procedure interface definitions for interacting with the SWSS.WORKER\_MOD package:

```
/*****
*      This procedure modifies data in the worker table.      *
*****/
```

PROCEDURE save\_worker\_name -- originally named modify\_worker\_table

```
(i_old_login_name  IN security.login_name%TYPE,
 i_new_login_name  IN security.login_name%TYPE,
 i_last_name       IN worker.last_name%TYPE,
 i_name_suffix     IN worker.name_suffix%TYPE,
 i_first_name      IN worker.first_name%TYPE,
 i_middle_name     IN worker.middle_name%TYPE,
 i_ssn             IN worker.ssn%TYPE,
 --i_security_code  IN security.security_code%TYPE,
 o_worker_id       OUT worker.worker_id%TYPE
);
```

```
/*****
*      This procedure modifies data in the load table.      *
*****/
```

PROCEDURE save\_worker\_load -- originally named modify\_load\_table

```
(i_old_load_no     IN load.load_no%TYPE,
 i_worker_id       IN load_worker.worker_id%TYPE,
 i_new_load_no     IN load.load_no%TYPE,
 i_provider_mgt_ind IN load.provider_mgt_ind%TYPE,
 i_supervisor_load_no IN load_worker.supv_load_no%TYPE,
 i_alternate_load_no IN load.load_no%TYPE,
 i_update_flag     IN load.provider_mgt_ind%TYPE,
 i_proj_ind        IN load.project_ind%TYPE
);
```

```
/*****
*      The modify_security_table procedure must be called if either the      *
*      worker's load number was changed or if the worker's security level was *
*      changed.                                                                *
*      Must now check for any load table records where old load number may   *
*      have been flagged as a supervisor. If any records found, that supervisor *
*      field will be cleared from worker records.                            *
*****/
```

PROCEDURE save\_security\_code -- originally named modify\_security\_table

```
(i_old_load_no     IN load.load_no%TYPE,
 i_new_load_no     IN load.load_no%TYPE,
 i_old_security_code IN security.security_code%TYPE,
 i_new_security_code IN security.security_code%TYPE,
 i_update_flag     IN load.provider_mgt_ind%TYPE,
 i_worker_id       IN security.worker_id%TYPE
);
```

PROCEDURE save\_worker\_phone -- originally named modify\_worker\_phone\_record

```
(i_worker_id  IN worker_phone.worker_id%TYPE,
```

July 06, 2001

---

```
    i_area_code    IN phone.area_code%TYPE,  
    i_phone_no     IN phone.phone_no%TYPE,  
    i_extension    IN worker_phone.extension%TYPE,  
    i_phone_type   IN worker_phone.phone_type_code%TYPE  
);
```

```
PROCEDURE save_worker_address    -- originally named modify_worker_address_record  
(i_worker_id      IN worker_address.worker_id%TYPE,  
 i_line1          IN address.line1%TYPE,  
 i_line2          IN address.line2%TYPE,  
 i_city           IN address.city%TYPE,  
 i_state_code     IN address.state_code%TYPE,  
 i_zip            IN address.zip%TYPE,  
 i_zip_plus       IN address.zip_plus%TYPE,  
 i_email_address  IN worker.email_address%TYPE,  
 i_address_type   IN worker_address.addr_type_code%TYPE,  
 i_office_addr_used IN VARCHAR2,  
 i_office_addr_id IN worker_address.address_id%TYPE  
);
```

```
PROCEDURE save_load_program    -- originally named modify_load_program_table  
(i_old_load_no IN load_program.load_no%TYPE,  
 i_new_load_no IN load_program.load_no%TYPE,  
 i_program_code IN load_program.program_code%TYPE,  
 i_active_ind  IN VARCHAR2  
);
```

```
/*****  
*      This procedure maintains the records in the worker_language table.      *  
*****/
```

```
PROCEDURE save_worker_languages    -- originally named modify_worker_language_table  
(i_worker_id      IN worker_language.worker_id%TYPE,  
 i_old_language_code IN worker_language.language_code%TYPE,  
 i_new_language_code IN worker_language.language_code%TYPE  
);
```

```
PROCEDURE add_load_to_worker  
(i_worker_id      IN load_worker.worker_id%TYPE,  
 i_new_load_no     IN load.load_no%TYPE,  
 i_provider_mgt_ind IN load.provider_mgt_ind%TYPE,  
 i_supervisor_load_no IN NUMBER,  
 i_alternate_load_no IN NUMBER  
);
```

```
PROCEDURE reassign_active_case  
(i_old_load  IN load_case.load_no%TYPE,  
 i_new_load  IN load_case.load_no%TYPE,  
 i_log_id    IN load_case.log_id%TYPE,  
 i_new_county IN load_case.county_no%TYPE  
);
```

```
/*****  
*      This procedure takes the passed load number and inactivates all records in      *  
*      all tables which use that load number (except the load_case table, which is      *  
*      taken care of in the reassign_active_case procedure). Because of Foreign Key      *  
*      constraints, the load number record cannot be deleted from the load table. This *  
*****/
```

July 06, 2001

```
*      means that the load table acts as a history keeper, somewhat like the      *
*      load_program, load_worker, and security tables.                          *
*****/
PROCEDURE inactivate_load_no      -- originally named inactivate_load_number
(i_load_number IN load_case.load_no%TYPE
);

/*****
*      i_task - 'END' Assume that passed record info exists in a record. This task      *
*      sets the record's end_date column to SYSDATE.                                *
*      If i_task is anything else, insert a new record if an active record matching      *
*      the passed one does not exist.                                                *
*****/
PROCEDURE save_security_assign      -- originally named modify_coord_table
(i_worker_id  IN co_security_coord.security_coord_id%TYPE,
 i_county_no  IN co_security_coord.county_no%TYPE,
 i_district_no IN co_security_coord.district_no%TYPE,
 i_task       IN VARCHAR2
);

/*****
*      This procedure first tries to modify any existing records in the      *
*      co_adoption_supvsr table with the i_co_no county number. If that fails,      *
*      a new record is inserted into the co_adoption_supvsr table.                *
*      Modified code to allow multiple "county adoption supervisors" in each      *
*      county. This was done to conform to Utilities User Requirement            *
*      UT-1.5.9.                                                                *
*****/
PROCEDURE save_ADPT_sup_co      -- originally named change_adpt_supervisor_co
(i_load_no IN co_adoption_supvsr.load_no%TYPE,
 i_co_no  IN co_adoption_supvsr.county_no%TYPE
);

/*****
*      This procedure deletes any record in the co_adoption_supvsr table      *
*      which matches the input parameters. This is the only procedure in this      *
*      package that deletes records.                                            *
*****/
PROCEDURE delete_ADPT_sup_co      -- originally named delete_adpt_supervisor_co
(i_load_no IN co_adoption_supvsr.load_no%TYPE,
 i_co_no  IN co_adoption_supvsr.county_no%TYPE
);

/*****
*      This procedure is used to update the "primary_worker_id" column of the      *
*      SWSS_CASE table to make sure that the most recent worker_id for the      *
*      passed load number's active cases is stored in this column. This          *
*      "synchronization" helps to support the "transferring" of a case load when      *
*      a load number assignemnt switches from one login username to another.      *
*****/
PROCEDURE sync_worker_ids      -- originally named synchronize_worker_ids
(i_load_no IN load_case.load_no%TYPE,
 i_worker_id IN swss_case.primary_worker_id%TYPE
);

/*****
```

July 06, 2001

```
*      The LOA2 worker does not store a login_name, so, we will use a modified *
*      version of the modify_worker_table procedure to get this poor LOA2 soul *
*      into the database. *
*****/
PROCEDURE save_LOA2_worker      -- originally named modify_LOA2_worker
(i_old_worker_id  IN worker.worker_id%TYPE,
 i_last_name      IN worker.last_name%TYPE,
 i_name_suffix    IN worker.name_suffix%TYPE,
 i_first_name     IN worker.first_name%TYPE,
 i_middle_name    IN worker.middle_name%TYPE,
 i_ssn           IN worker.ssn%TYPE,
 o_worker_id      OUT worker.worker_id%TYPE
);

/*****
*      Created to fulfill LOA2 requirements for SWSS Utilities. *
*****/
PROCEDURE save_worker_hrs      -- originally named store_worker_hrs
(
 i_worker_id IN loa.worker_hours.worker_id%TYPE,
 i_load_no  IN loa.worker_hours.load_no%TYPE,
 i_hours1   IN loa.worker_hours.availability1%TYPE,
 i_hours2   IN loa.worker_hours.availability2%TYPE
);

/*****
*      Created to try and incorporate the saving of all SWSS profile data into a *
*      single stored procedure. This one just saves a new record. *
*****/
PROCEDURE save_new_profile
(i_login_name      IN security.login_name%TYPE,
 i_password        IN VARCHAR2,
 i_first_name      IN worker.first_name%TYPE,
 i_last_name       IN worker.last_name%TYPE,
 i_name_suffix     IN worker.name_suffix%TYPE,
 i_middle_name     IN worker.middle_name%TYPE,
 i_ssn            IN worker.ssn%TYPE,
 i_security_code   IN security.security_code%TYPE,
 i_load_no        IN load.load_no%TYPE,
 i_program_access  IN VARCHAR2,
 i_supervisor_load_no IN load_worker.supv_load_no%TYPE,
 i_alternate_load_no IN load.load_no%TYPE,
 i_proj_ind       IN load.project_ind%TYPE,
 i_create_account  IN VARCHAR2,
 i_skip_worker_security IN VARCHAR2,          --flag for skipping worker and
                                              --security table changes.
 i_worker_id      IN worker.worker_id%TYPE,   --Is only used if
                                              --i_skip_worker_security = 'T'
 o_worker_id      OUT worker.worker_id%TYPE
);

/*****
*      Created to check for an active load number match to the load number *
*      the user is trying to assign to a profile. This procedure will probably be *
*      called after a login_name record has already been detected for a *
*      proposed new username. *
*****
```

\*\*\*\*\* /  
PROCEDURE CheckLoadNoMatch

```
(i_login_name      IN load_worker.login_name%TYPE,  
i_load_no         IN load_worker.load_no%TYPE,  
i_worker_table_change IN VARCHAR2,  
i_first_name      IN worker.first_name%TYPE,  
i_middle_name     IN worker.middle_name%TYPE,  
i_last_name       IN worker.last_name%TYPE,  
i_name_suffix     IN worker.name_suffix%TYPE,  
i_ssn            IN worker.ssn%TYPE,  
i_security_code   IN security.security_code%TYPE,  
o_EvrythingOK     OUT VARCHAR2);
```

PROCEDURE ConvertLOA\_Load

```
(i_load_no      IN load_worker.load_no%TYPE,  
i_login_name   IN load_worker.login_name%TYPE,  
i_password     IN VARCHAR2,  
i_create_account IN VARCHAR2,  
i_program_access IN VARCHAR2,  
i_first_name   IN worker.first_name%TYPE,  
i_last_name    IN worker.last_name%TYPE,  
i_name_suffix  IN worker.name_suffix%TYPE,  
i_middle_name  IN worker.middle_name%TYPE,  
i_ssn         IN worker.ssn%TYPE,  
i_security_code IN security.security_code%TYPE,  
i_supervisor_load_no IN load_worker.supv_load_no%TYPE,  
i_alternate_load_no IN load.load_no%TYPE,  
o_worker_id    OUT worker.worker_id%TYPE);
```

## Data Warehouse

Most of the data from the following tables may be stored in the data warehouse:

- Load
- Load\_worker
- Security
- Worker
- Worker\_name\_hist

If any other Utilities-related data is stored in the data warehouse, I am unaware of it.

## Technical Issues

Discuss any tricky things that happen in the module that someone who maintains the application may not recognize at first glance. Sibling group sharing, legal status switches, or reusing person IDs.

User requirement UT-1.4.3: Provide a mechanism that allows a user to have both a SWSS profile and a LOA2 profile. Only allow this type of profiling from the SWSS Application mode. If the load number detection procedure detects that the user already has an LOA2 load number, ask if this person is the same as the profile being worked on. If the user says yes, make the load number a BOTH. If the user says no, then give the user a warning message and prevent whatever modification is taking place.

Alternate Workers. The issue of how alternate worker assignments are read, displayed, and stored is a tricky one. Here is a rough draft of a memo I created which should help explain how it works to future developers of this project.

July 06, 2001

---

STATE OF MICHIGAN  
**FAMILY INDEPENDENCE AGENCY**

---

---

**MEMORANDUM**

---

To:

Date:

From: Matthew D. Miller

Subject: Alternate Worker to Worker relationship (VERY ROUGH DRAFT)

These are the only requirements I could find that deal with Alternate workers.

- UT-1.5.5            Provide multiple mechanisms for quickly assigning an Alternate staff profile to the current staff profile (details follow).
- UT-1.5.5.1        Pull down the list by selecting the down arrow, scroll to the appropriate entry, and then select that entry.
- UT-1.5.5.2        Select the Alternate field, then type up to five characters to “search” for the appropriate entry; i.e., a locate function.
- UT-1.5.5.3        Provide a mechanism (such as using the delete key) to remove the currently selected Alternate profile from the current profile.
- MM-1.1.2.2.4.3    The user’s relationship to the selected case. This covers the primary worker assigned to the case, the alternate worker assigned to the load number of the primary worker of the case, the supervisor of the primary worker of the case, *et cetera*.<sup>1</sup>

The SWSS.UTILITIES.Get\_Worker\_Info procedure is the first one used to gather profile information. The load number that the user selected from the Worker list as well as the county number (taken from that load number) is passed as parameters into the procedure. The procedure populates a REF CURSOR from many different sources. The basic source is a record in the load\_worker table that matches the load\_no column with the passed load number and has an end\_date value of NULL. The load\_no value from this record is then used to get the correct record from the load table. The load table record provides the correct provider\_mgt\_ind column value as well as the alt\_worker\_id column value. The alt\_worker\_id value (and the county number procedure parameter) is then passed to a function called SWSS.UTILITIES.Worker\_LoadNo.

The SWSS.UTILITIES.Worker\_LoadNo function then gathers the highest numbered load number actively assigned to the alt\_worker\_id for the passed county. Both the load\_worker and load tables are used to gather this unique record. Just as in the SWSS.UTILITIES.Get\_Worker\_Info procedure, an active load number

---

<sup>1</sup> This can also be called the user’s responsibility for the selected case.

assignment is designated by a load\_worker record with a NULL value in the end\_date column. This “MAX” load number is the Alternate Worker load number passed back to Visual Basic and displayed in the Alternate Worker field for the profile selected by the SWSS user.

That takes care of how the Alternate Worker load number assignment is determined for a particular profile. Now to how Alternate Worker ID is stored into the load table for a profile’s load number.

The SWSS.WORKER\_MOD.AdminLoadTable function as well as the SWSS.WORKER\_MOD.save\_worker\_load procedure both store an alt\_worker\_id into the appropriate load table record in the same way. Both procedure receive an Alternate Worker load number as a parameter. This load number was selected from the Alternate Worker field on Screen 2 of the SWSS Utilities application. This load number is then “translated” into a worker\_id by using a SELECT statement that locates a record in the load\_worker table. The load\_worker table record will have a load\_no column that matches the passed Alternate load number and an end\_date column value of NULL. The worker\_id column value for that record is then stored into the alt\_worker\_id column of the load table.

In conclusion, it can be seen that, based upon the database designs as well as the SQL code that has been written, Alternate workers are assigned to a profile’s load number, not to a profile’s worker\_id/username.

However, that conclusion is not the end of the story. Now that we see how the SWSS profile is set up, now we have to take that profile setup information and see how it is used in relation to Case Security.

When a user selects a case from the Soundex, Case Listing, or Main Menu module, the Main Menu module must eventually determine a list of users who have access to the selected case. The Main Menu is also the first step in determining how much access a user is given to a case (however, it is not the only step in deciding case access).

The principal procedure for determining who has access to a case is the SWSS.Main\_Menu.get\_update\_worker\_ids procedure. This procedure has only one input parameter, the “primary” load number of the selected case. That load number is determined by the load\_case table record where the log\_id column matches the log number of the selected case, the county\_no column matches the county number found in the swss\_case table record for the selected log number, the responsibility column has a value of ‘P’ (for “primary”), and the value in the end\_date column is NULL. Here is the query which is used to gather a list of all worker\_ids that will have full update access to a selected case.



```
SELECT    worker_id    o_worker_ids
FROM      load_worker
WHERE     end_date IS NULL
connect by prior supv_load_no = load_no
start with load_no          = i_primary_load_number
union
SELECT alt_worker_id o_worker_ids
FROM load
WHERE load_no in
      (SELECT load_no
       FROM load_worker
       WHERE end_date IS NULL
       CONNECT BY prior supv_load_no = load_no
       START WITH load_no          = i_primary_load_number);
```

By looking at this complex query, we can still see that determining the Alternate to Worker profile relationship is still based upon load number. However, because the result of this query is a list of worker\_ids, it then becomes evident that the Alternate to Worker relationship is eventually “translated” into a relationship between worker\_ids.

In a final conclusion, we can make the following observation. The database design stores the Alternate to Worker relationship based upon load number. However, because the SWSS application needs to identify the current SWSS user based upon his/her worker\_id, this load number relationship MUST eventually be “translated” into a worker\_id relationship. It is the only way for the SWSS application to positively identify who the current SWSS user is (or who the user *claims* to be). Based upon this identification and a lot of other information, the SWSS application can then determine how much access to case information and SWSS functionality the user should be granted.

<<End of Memo>>

## Test Plans

## Test Plan – SWSS Utilities

Matthew D. Miller

### Process Accessibility

- All valid SWSS workers are allowed, at least, some level of access to Utilities. A case does **not** have to be selected in order to execute the Utilities process.
- Central Office Security Coordinators and Local Office Security Officers are the only types of workers allowed access to the Add Staff Profile and Inactivate Staff Profile functions. All SWSS workers are allowed access to the Ticklers, Change Staff Profile, and Reconciliation functions.
- Non-security coordinators may only change passwords for their own worker records. They must correctly specify their current (or “Old”) password and enter their new password twice before any password changes will be applied to their Oracle account.
- Central Office Security Coordinators may change all other worker data fields on all four screens for all workers in the State Of Michigan. However, when they change another worker’s password, they will **not** be required to enter a current password. They will just need to correctly enter the new password twice. Also, Central Office Security Coordinator worker records are not accessible or visible from SWSS Utilities. Their records are maintained outside of the system. Central Office Security Coordinators are also the only workers allowed to create Local Office Security Coordinators.
- Local Office Security Coordinators are assigned county-district combinations that limit which worker records they are allowed to maintain. The only worker records they will see will be workers who are assigned an active load number, which contains one of the county-district combinations, assigned to the coordinator. The Local Office Security Coordinator may then change any worker data fields on all four screens for all of these workers, except their own record. The only data a coordinator may change in his/her own record is his/her password. Coordinators must specify their current password when changing their own record; otherwise they only need to enter a new password twice when modifying other worker records. On the fourth screen, Local Office Security Coordinators may **not** change the Security Administration check box or any county-district combinations for any worker records.
- If the worker being modified is assigned to the Adoption program and given a Supervisor or higher security level, it is possible for a Central/Local Office Security Coordinator to assign a list of counties to the selected worker. These counties represent which counties the worker “covers” for Adoption Supervision.
- Corrections Mode does not apply to the Utilities process.
- If an existing worker’s name fields are changed, the original name is written to a worker name history database table (worker\_name\_hist). There is currently no mechanism for viewing this name history.

- When changing a worker's load number, a prompt should come up asking if the new load is to be added to the worker's profile, or if the new load number is meant to replace the current load number.
  - If the former, the load number should become a new valid load number for the worker.
  - If the latter, the database is scanned for active cases attached to the old load.
    - If any cases are found, the user is asked to assign different active load numbers to those cases. Otherwise, the load number change is made.
  - Carefully note that adding any new load numbers to a workers profile is meant to be a part of the Change Staff Profile function, **not** the Add New Staff Profile function.
- Maximum field lengths:
  - Last name = 19 characters
  - First name = 12 characters
  - Middle = 1 character
  - Suffix = 3 characters
  - Username = 30 characters
  - Password = 30 characters (with a minimum of 6 characters)
  - Section, Unit and Worker numbers are limited to two digits
  - Address Line 1 = 60 characters
  - Address Line 2 = 60 characters
  - City = 30 characters
  - Zipcode = 5 digits
  - Zip Plus = 4 digits
  - Phone number = 10 digits
  - Phone extension = 5 digits
- The Inactivate Staff Profile function is first used to inactivate load numbers assigned to a worker (especially if the worker has been assigned multiple load numbers). It is not until the worker's last active load number is inactivated that a worker will be denied access to the SWSS application.
- The user should not be allowed to Inactivate a worker load number if any active case is still attached to it. If a worker's last load number is inactivated, the user should be asked if they wish to eliminate the worker's Oracle account.

#### Case Functionality

- A valid SWSS case must be selected before Reconciliation may be executed. If one hasn't been selected, user should be prompted to "jump" to the Case Listing process for case selection.
- Only Juvenile Justice, Foster Care, and Adoption cases should be allowed to be reconciled. Also, those cases must have a status of either Unregistered or Registered.

More Pertinent Information

Staff Profile Properties

Property Name	Constraints	
SSN (Social Security Number)	Format ###-##-#### (10 character string)	
Project Indicator	Valid Values: S = SWSS L = LOA2 B = Both SWSS and LOA2	
(Name) Last	Max 30 characters	
(Name) First	Max 30 characters	
(Name) Middle Initial	Max 1 character	
(Name) Suffix	Max 3 characters	
(Worker Access) ADPT (Adoption)	Boolean	
(Worker Access) CFC (Child Foster Care)		
(Worker Access) CPS (Child Protective Services)		
(Worker Access) JJ (Juvenile Justice)		
(Worker Access) Provider Management		
(Worker Access) Security Administration		
(Mailing Address) City		
(Mailing Address) Line 1		
(Mailing Address) Line 2		
(Mailing Address) Zip Code		
(Mailing Address) Zip Plus		
(Primary Address) City		
(Primary Address) Line 1		
(Primary Address) Line 2		
(Primary Address) Zip Code		
(Primary Address) Zip Plus		
County Adoption Supervisor's list of counties		
Security Administrator's list of county-district pairs.		
E-Mail Address		
(Phone) Fax Number		
(Phone) Fax Extension		
(Phone) Primary Number		
(Phone) Primary Extension		
Availability Hours 1		
Availability Hours 2		
Language 1		
Language 2		
Language 3		
Alternate Worker's Load Number (worker_id)		
Supervisor's Load Number		
(Load Number) County Number		
(Load Number) District Number		
(Load Number) Section Number		
(Load Number) Unit Number		
(Load Number) Worker Number		
Password		

July 06, 2001

Username		
Oracle Role(s)		
Security Level		
Worker ID		

- The Utilities process first must determine if Utilities is already running. If a Previous Instance of Utilities is found, shut down the new instance and set focus to the once that is already running.
- Run the ExtractINI\_Info procedure to gather information about the user who is logged into the system. Information that will be used from this file by the Utilities process is this:
  - [Current User]
    - User Name. (gsUserName)
      - Used to connect to the Oracle database.
    - Password. (gsPasswd)
      - Used to connect to the Oracle database.
      - Used to allow the user to change his/her own password.
    - Program(s) (gtPrograms)
      - Used to determine if the user has access to the following Utilities Actions:
        - Add Staff Profile
        - Change Staff Profile
        - Inactivate Staff Profile
        - Display Inactive Load Numbers
    - Security Level (gtUserSecurityLevel)
      - Used to determine if the user has access to the Payment Document Control Utility action.
  - [Other]
    - DB Connect String (gtDB\_Connection)
      - Determines which database the user is connected to.
    - Central Office Code (gtCentOfficeCode)
      - The two-digit number which refers to the Central Office (as if it was its own county).
  - [Case Info]
    - Log Number. (gtLogNumber)
      - Along with Case Status, determines access to the CPS Transfer Information action.
    - Case Status (gtCaseStatus and gtCaseState)
      - Along with Log Number, determines access to the CPS Transfer Information action.
  - Determine who the user is. Here are the classifications of user that the Utilities module might possibly expect:
    - Central Office Security Coordinator
      - If the gtPrograms variable contains the string "SC84" (where 84 is the value of the variable gtCentOfficeCode), then the user must definitely be a Central Office Security Coordinator.
      - The value of gtLogNumber should also be <= 0.
    - Local Office Security Coordinator

July 06, 2001

- If the gtPrograms variable contains the string "SEC", then the user must definitely be a Local Office Security Coordinator.
- The value of gtLogNumber should also be <= 0.
- PROGRAM OFFICE OR P/DC user
  - If the gtUserSecurityLevel = 80, then the user is a PROGRAM OFFICE OR P/DC user.
  - The value of gtLogNumber should also be <= 0.
- All other users. If the current user does not fall within the other previous categories, then this user can simply be considered an Other User.
- This list should cover every possible user. If the current user falls out of this list, then Utilities does not know what to do with him/her.
- Determine which Actions the user has access to.

<b>SWSS Utilities Action Access Chart</b>				
<b>User →</b>	Central Office Security Coordinator	Local Office Security Coordinator	PROGRAM OFFICE OR P/DC user	Other User
<b>Action ↓</b>				
Add Staff Profile	X	X		
Change Staff Profile	X	X	X	X
Inactivate Staff Profile	X	X		
Display Inactive Load Numbers	X	X		
CPS Transfer Information				If a log number was previously selected and gtCaseState = 400 (Unregistered)
Managing Related Cases				If gtCaseState <> 440 and gtCaseState <> 430
Payment Document Control Utility			X	

**X** = User has access to action.

**Blank** = User does not have access to action.

- Display the Action Panel.
- Position Profile Data Panels A, B, C, and D. Also, keep them invisible.
- Hide the Cancel/Previous and Next/Continue buttons.
- Center the Action panel and the Central Office Security Password frame onto the form. The password frame starts out invisible, however, the Action panel is visible and accessible.
- Resize all controls (and "fields") to be proportionate with the user's chosen video resolution.
- Determine which menu items the user has access to.

<b>SWSS Utilities Menu Access Chart</b>				
<b>User →</b>	Central Office Security Coordinator	Local Office Security Coordinator	PROGRAM OFFICE OR P/DC user	Other User
<b>Menu ↓</b>				
File	X	X	X	X
Save				
Print Screen	X	X	X	X
Exit	X	X	X	X
Sections				X
Main Menu				X
Case Listing				X
Case Registration				If (Length of gtLogNumber > 0) And (gtCaseState = 400)
Child Information				If (Length of gtLogNumber

July 06, 2001

<b>SWSS Utilities Menu Access Chart</b>				
<b>User →</b> <b>Menu ↓</b>	Central Office Security Coordinator	Local Office Security Coordinator	PROGRAM OFFICE OR P/DC user	Other User
				> 0) And (Length of gtProgramCode > 0) And (gtCaseState <> 400) And (gtCaseState <> 425)
Member Information				Same as Child Info
Legal				Same as Child Info
Funding Determination				(Same as Child Info) And If (gtProgramCode = CFC)
Placement				Same as Child Info
Payment				Same as Child Info
Education				Same as Child Info
Medicaid				Same as Child Info
Medical Passport				
Case Closing				
Five Day Packet/Placement Outline				Same as Child Info
Comments				Same as Child Info
Case Summary				If (Length of gtLogNumber > 0) And (gtCaseState <> 425)
MARE Registration				(Same as Child Info) And If (gtProgramCode = ADPT)
Ticklers				X
Provider				X
Report Generation				X
Utilities				
Autoflow				X
Autoflow Enabled				X
Corrections				
Corrections Mode				
View				
Help	X	X	X	X
Contents	X	X	X	X
Search For Help On...	X	X	X	X
How to Use Help	X	X	X	X
About Utilities Section...	X	X	X	X
Password	X <sup>2</sup>			

**X** = User has access to menu.

**Blank** = User does not have access to menu.

- Determine which “Worker Update Access” checkboxes the user has access to.
  - If the length of gtPrograms <= 1 then, the user is denied access to all Worker Update Access checkboxes. Also, when the user is performing the “Inactivate Staff Profile” action, all “Worker Update Access” Checkboxes are disabled. Otherwise, it follows the following chart.

<sup>2</sup> If the username is CENT\_ADM, this menu item will remain invisible.

July 06, 2001

<b>Chart for Determining Access to “Worker Update Access” Checkboxes</b>				
<b>User →</b>	Central Office Security Coordinator	Local Office Security Coordinator	PROGRAM OFFICE OR P/DC user	Other User
<b>“Worker Update Access” checkbox ↓</b>				
Protective Services	X	X		
Child Foster Care	X	X		
Adoption	X	X		
Juvenile Justice	X	X		
Provider Management	X	X		
Security Administration	X			

**X** = User has access to “Worker Update Access” checkbox.

**Blank** = User does not have access to “Worker Update Access” checkbox.

*Note: The Local Office Security Coordinator will be denied access to all “Worker Update Access” Checkboxes when viewing his/ her own profile or the profile of another Local Office Security Coordinator.*

- Gather a list of all county-district combinations that the current user has access to.
  - Load up the cboActionCounty control with all of the counties that the user has access to.
  - An extra entry, called “All Counties” is loaded into the cboActionCounty control for Central Office users. The ItemData property value for this entry is 1000, instead of the county code value used by all other entries.